

# (12) UK Patent Application (19) GB (11) 2 388 502 (13) A

(43) Date of A Publication 12.11.2003

(21) Application No: 0210704.3

(22) Date of Filing: 10.05.2002

(71) Applicant(s):  
Chris Dunn  
Walnut Tree Cottage, Hampton Poyle,  
Oxon, OX5 2QQ, United Kingdom

(72) Inventor(s):  
Chris Dunn

(74) Agent and/or Address for Service:  
Mathys & Squire  
100 Grays Inn Road, LONDON, WC1X 8AL,  
United Kingdom

(51) INT CL<sup>7</sup>:  
H04B 1/66 // H03M 7/46, H04L 29/06

(52) UK CL (Edition V):  
H4P PDCFX

(56) Documents Cited:  
EP 0884850 A2 WO 2001/062009 A1  
WO 1998/037700 A1

(58) Field of Search:  
UK CL (Edition T) H4F FRR FRT, H4P PDCFR  
PDCFX, H4R RPBE  
INT CL<sup>7</sup> G06F 17/14, H03M 7/46, H04B 1/66, H04L  
29/06, H04N 1/41 5/926 7/30 7/50 9/804  
Other: Online: EPODOC, JAPIO, WPI

(54) Abstract Title: Compression of frequency domain audio signals

(57) Audio signals are transformed into frequency domain representation and then quantized. The coefficients can be thought of as a bar graph of magnitude against frequency (see figure).

The compression method involves considering the most significant bit plane and identifying the coefficients whose magnitude equals/exceeds a threshold. The positions of these coefficients are run length encoded and then removed from the coefficient list.

The process is then repeated for the next most significant bit plane and so on.

Coding of less significant bits of coefficients removed at more significant bit planes can also be performed.

Clustering of same frequency coefficients prior to encoding can improve the compression efficiency. The scheme can be applied to full bandwidth and layered bitplane encoding.

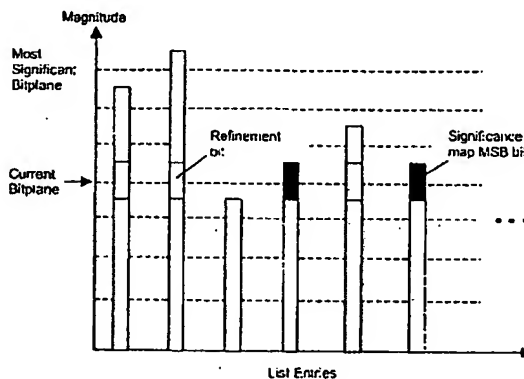


Figure 5

GB 2 388 502 A

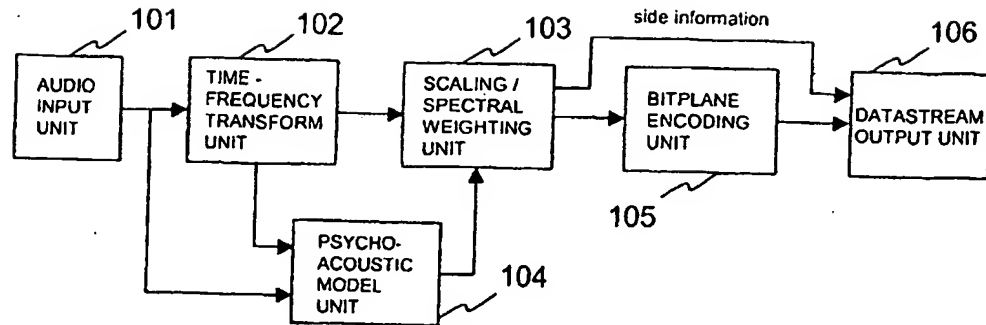


Figure 1

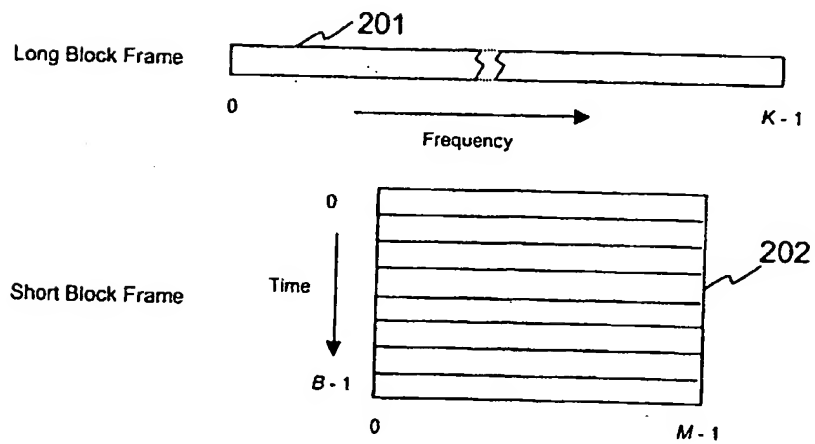


Figure 2

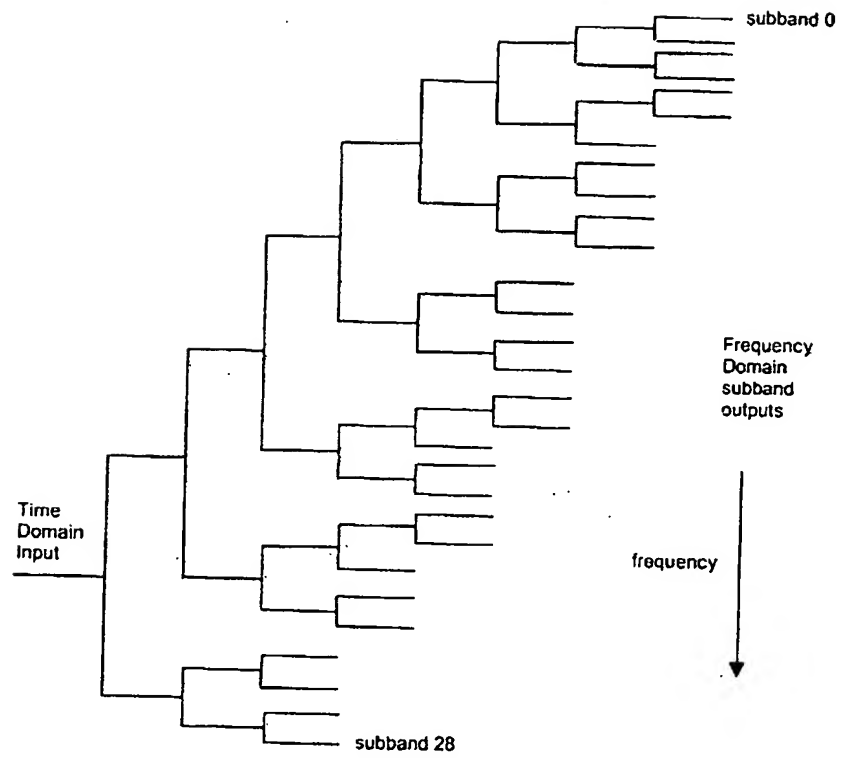


Figure 3

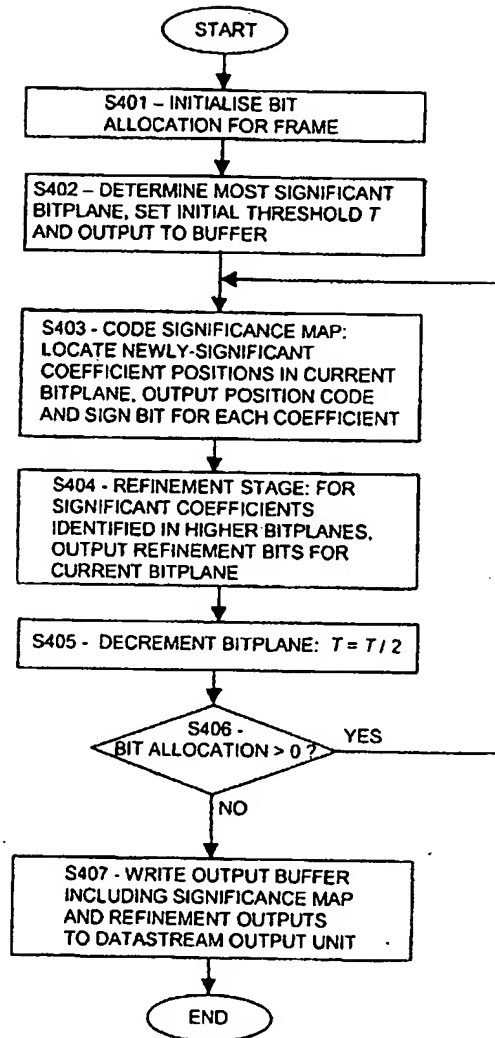


Figure 4

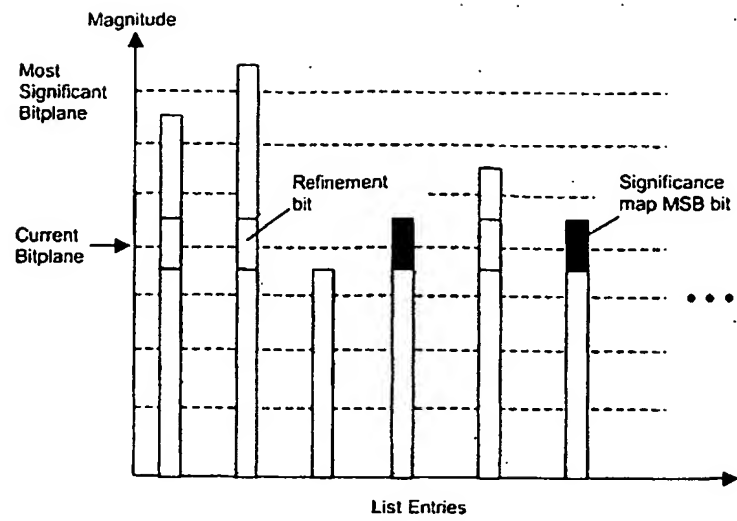


Figure 5

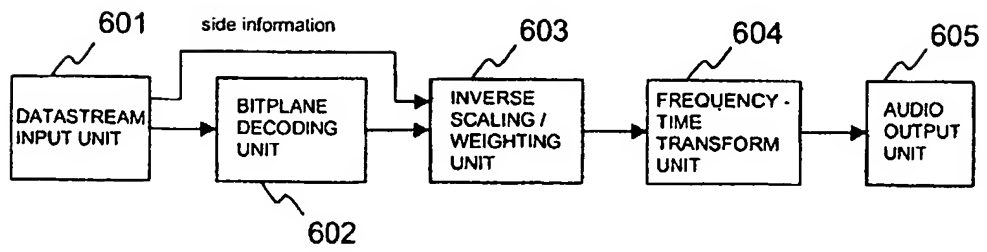


Figure 6

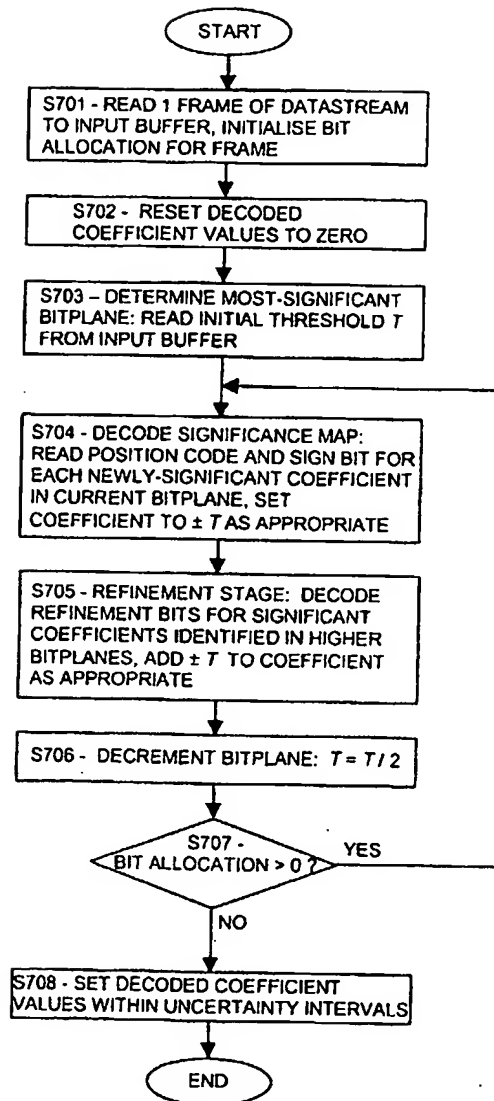


Figure 7

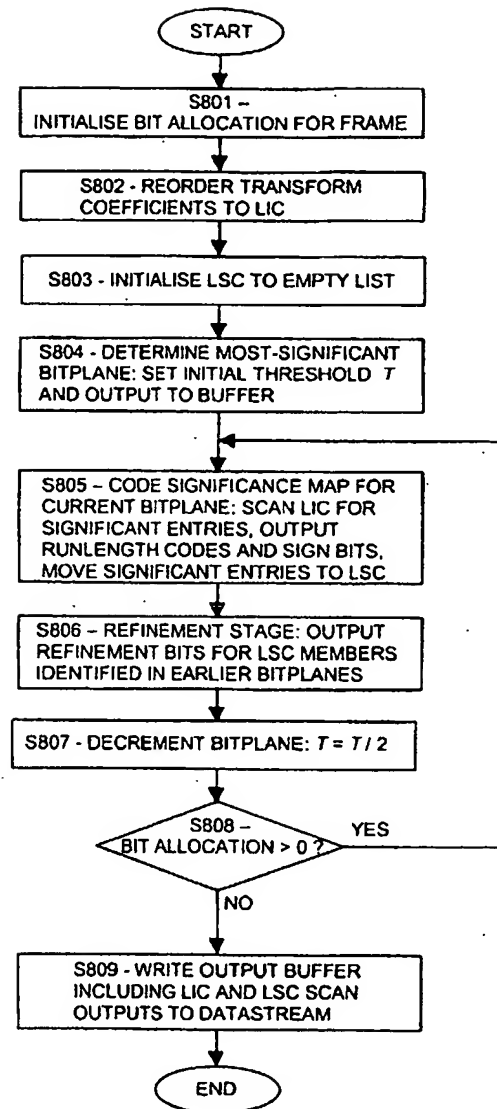


Figure 8

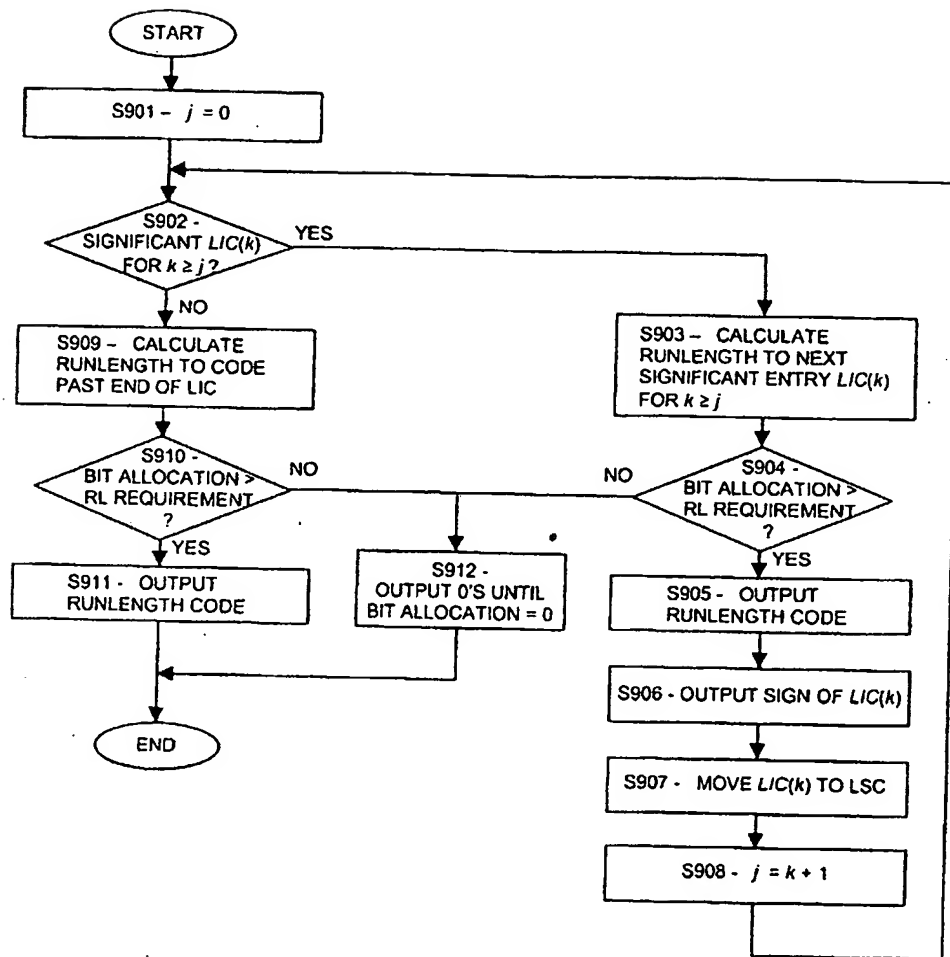


Figure 9

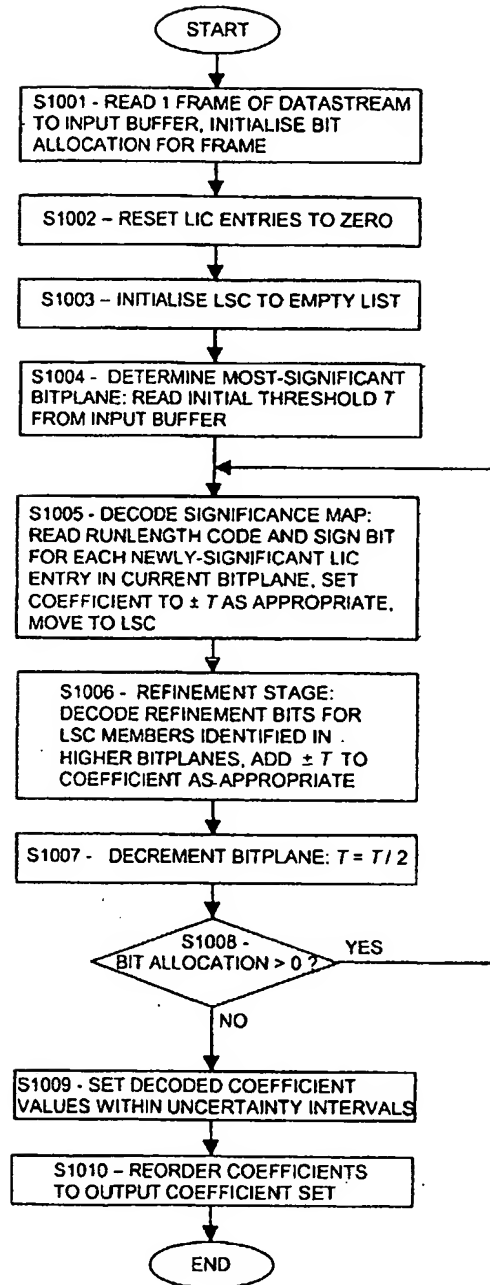


Figure 10

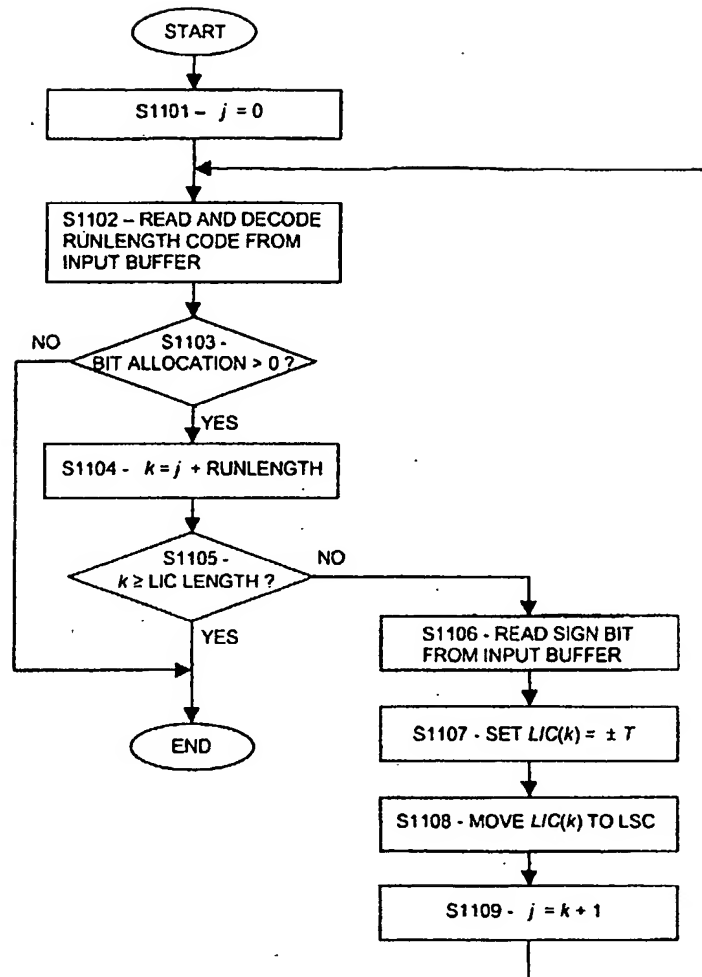


Figure 11

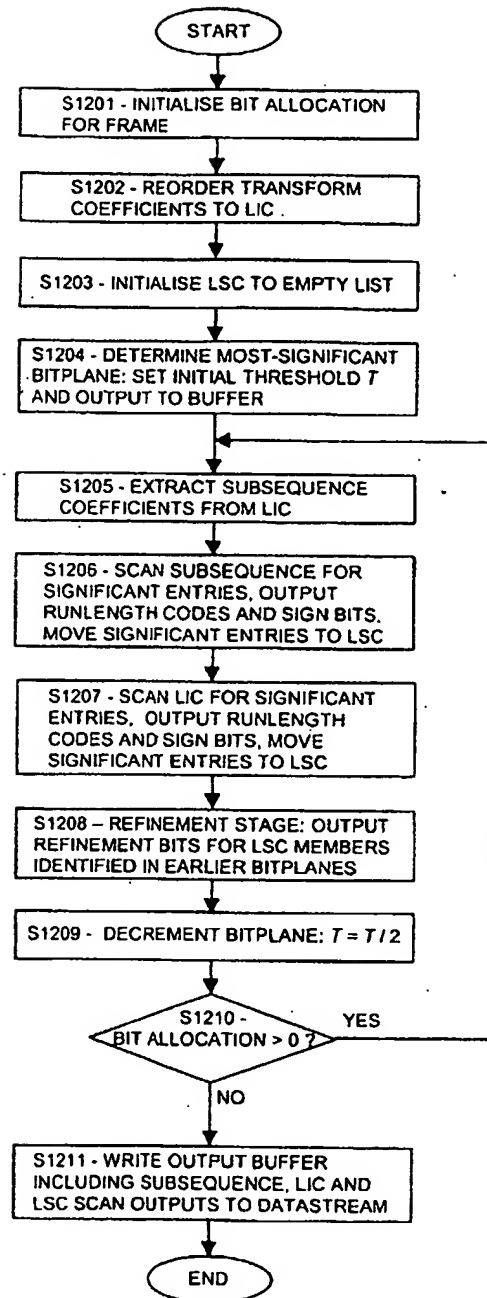


Figure 12

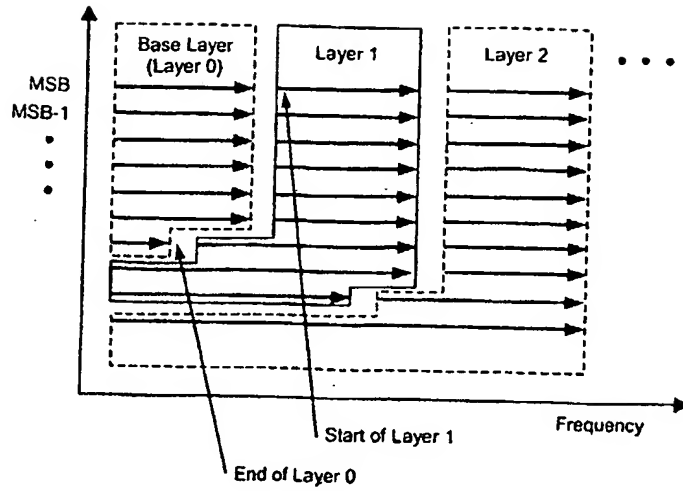


Figure 13

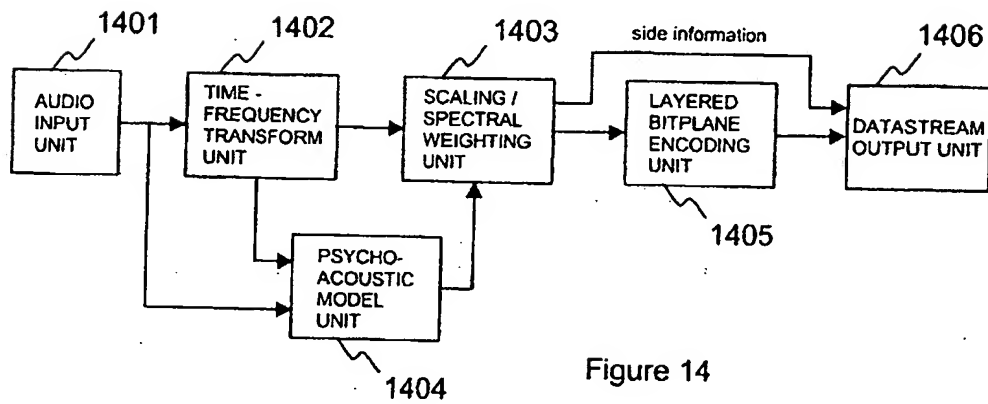


Figure 14

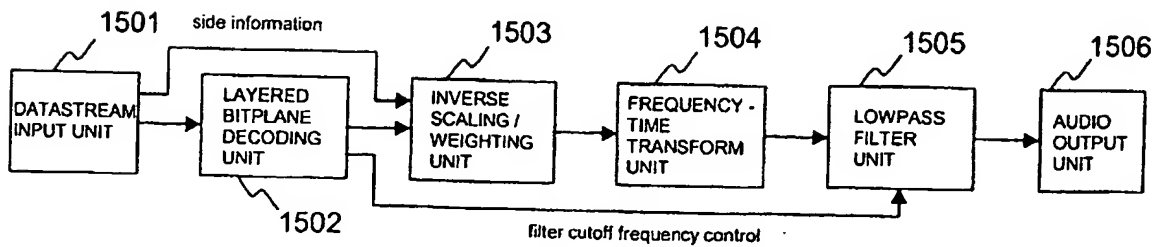


Figure 15

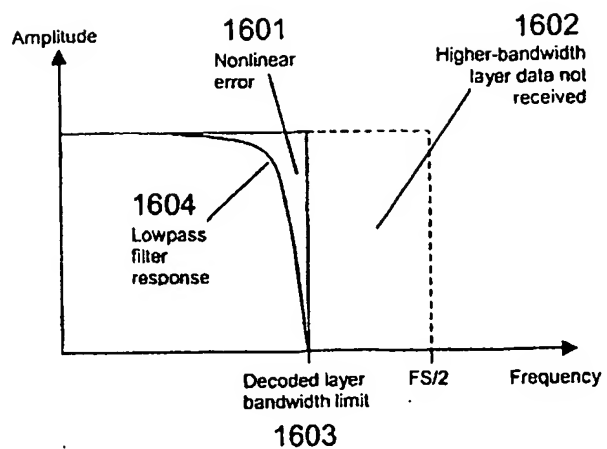


Figure 16

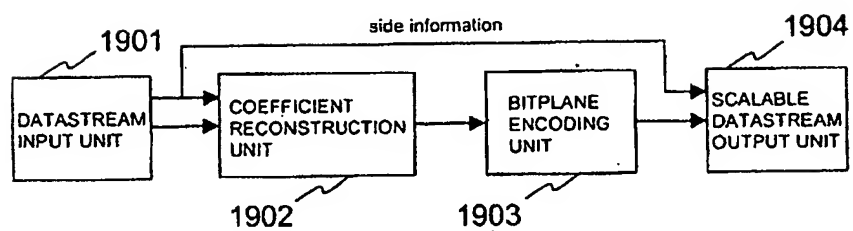


Figure 19

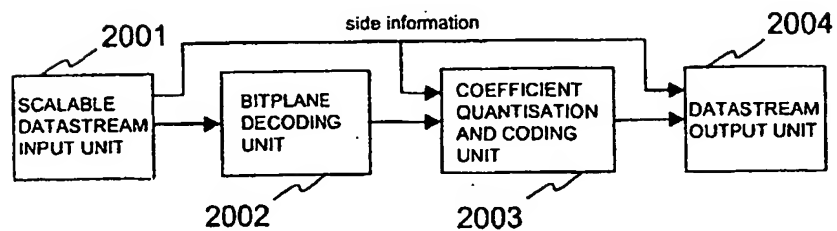


Figure 20

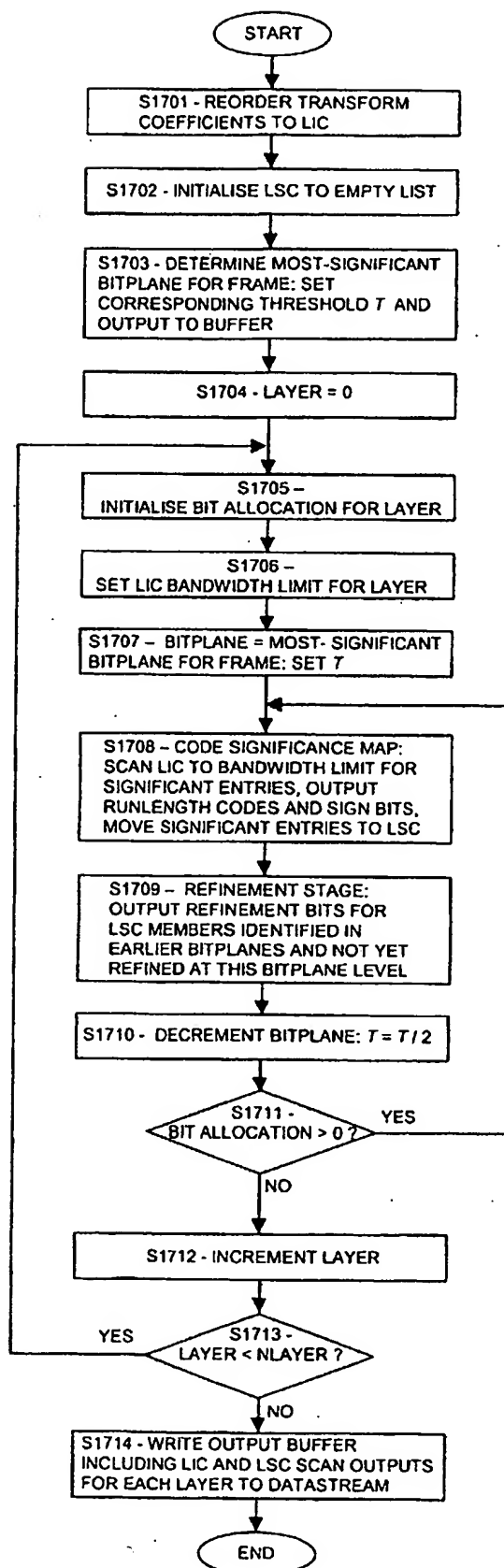


Figure 17

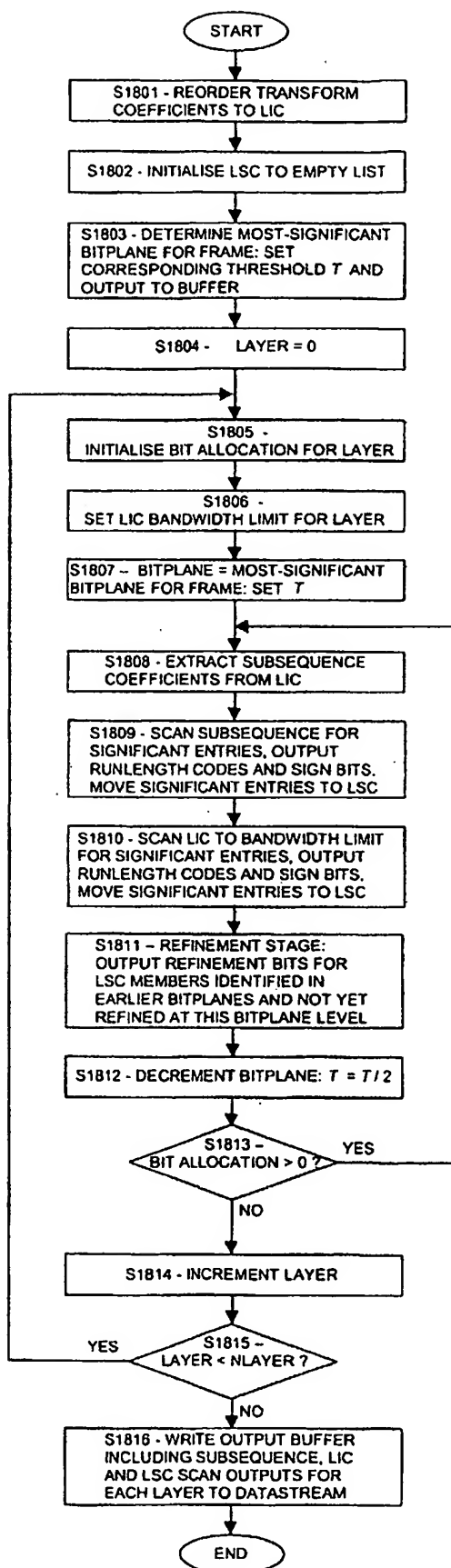


Figure 18

## Audio Compression

### Field of the Invention

- 5 This invention relates generally to the field of audio compression, in particular to efficient methods for encoding and scalably decoding audio signals.

### Background

- 10 Audio coding algorithms with bitrate scalability allow an encoder to transmit or store compressed data at a relatively high bitrate and decoders to successfully decode a lower-rate datastream contained within the high-rate code. For example, an encoder might transmit at 128 kbit/s while a decoder  
15 would decode at 32, 64, 96 or 128 kbit/s according to channel bandwidth, decoder complexity and quality requirements. Scalability is becoming an important aspect of low bitrate audio coding, particularly for multimedia applications where a range of coding bitrates may be required, or where bitrate fluctuates. Fine-grain scalability, where useful increases in coding  
20 quality can be achieved with small increments in bitrate, is particularly desirable.

- The growth of the internet has created a demand for high-quality streamed audio content. Audio coding with fine-grain bitrate scalability allows  
25 uninterrupted service in the presence of channel congestion, achieves real-time streaming with low buffer delay, and yields the most efficient use of available channel bandwidth. Scalability is also useful in archiving, where a program item may be coded at the highest bitrate required and stored as a single file, rather than storing many coded versions across the range of  
30 required bitrates. As well as the saving in overall storage requirement, bitrate scalability avoids the cumulative reduction in coding quality that can occur due to recoding. Scalable audio coding has further applications in

mobile multimedia communication, digital audio broadcasting, and remote personal media storage.

5 While fine-grain bitrate scalability can be extremely useful, it is important that it is achieved without significant coding efficiency penalty relative to fixed bitrate systems, and with low computational complexity.

10 Audio compression algorithms typically include some form of transform coding where the time-domain audio signal is split into a series of frames, each of which is then transformed to the frequency domain before quantisation, entropy coding and frame packing to a coded datastream. A psychoacoustic model determines a target noise shaping profile which is used to allocate bits to the transform coefficients such that quantisation errors for each frame are least audible to the human ear. In a conventional  
15 fixed-bitrate encoder the bit allocation is typically achieved with a recursive algorithm that attempts to meet the noise-shaping requirement within the bitrate constraint (see J. D. Johnston, "Transform Coding of Audio Signals Using Perceptual Noise Criteria," *IEEE J. Select Areas in Communications*, vol. 6, pp. 314 – 323 (1988 Feb.)). The final bit allocation computed is used  
20 to quantise transform coefficients and also included as side information within the datastream for use at the decoder. Datastream decoding is restricted to the bitrate of the encoded signal.

25 A common approach to achieving scalability is the 'error-feedforward' arrangement, (for example J. Herre et al., "The Integrated Filterbank Based Scalable MPEG-4 Audio Coder," presented at the 105<sup>th</sup> Convention of the Audio Engineering Society, San Francisco, 1998 (preprint 4810)), where a core coder produces the lowest embedded bit rate and subsequent layers progressively reduce the error due to the core. However, a significant  
30 amount of side information is associated with each layer which can reduce coding efficiency, and the number of possible decoding rates is limited to the number of layers.

An alternative approach to achieving scalability is ordered bitplane coding of transform coefficients, where in each frame coefficient bitplanes are coded in order of significance, beginning with the most significant bits (MSB's) and progressing to the least-significant bits (LSB's). This results in fully-embedded coding where the datastream at a certain rate contains all lower-rate codes, and exhibits fine-grain scalability in contrast to the coarse granularity offered by error-feedforward systems. A lower bitrate version of a coded signal can be simply constructed by discarding the later bits of each coded frame. Bitplane coding can also yield a significant increase in encoding speed since ordered bitplanes are coded sequentially until the bit allocation for the frame is met, as opposed to the recursive bit allocation search executed in fixed-rate coding.

Ordered bitplane coding is used in the Bit-Sliced Arithmetic Coding (BSAC) system (S. H. Park et al., "Multi-Layer Bit-Sliced Bit Rate Scalable Audio Coding," presented at the 103<sup>rd</sup> Convention of the Audio Engineering Society, New York, Sep. 1997 (preprint 4520), and S. H. Park, "Scalable Audio Coding / Decoding Method and Apparatus," EP 0884850 (1998 Dec.). However the BSAC coder requires the use of arithmetic coding which can increase computational complexity.

An object of this invention is to provide a method and apparatus for efficiently coding audio signals with fine-grain bitrate scalability.

### Summary of the Invention

According to the present invention there is provided a method for encoding audio signals to a datastream, comprising the steps of:

- (a) reordering frequency-domain coefficients representing the audio signal to a coefficient list, where the list order preserves the frequency order of

coefficients and groups together coefficients with the same frequency index;

5 (b) quantising the coefficients and coding bits of equal significance together in bitplanes, where bitplanes are coded in order of significance beginning with the most significant bitplane, and coding of one or more bitplanes comprises the steps of:

10 (i) locating newly-significant coefficients with most-significant magnitude bit (MSB) positions within the current bitplane, by runlength coding positions of coefficient list entries whose magnitudes equal or exceed a predetermined threshold level corresponding to the current bitplane;

15 (ii) coding the signs of said newly-significant coefficients;

(iii) removing said newly-significant coefficients from the coefficient list.

20 (c) outputting coded bitplane data to the datastream.

25 The datastream may comprise a base layer and a number of enhancement layers having predetermined bandwidth limits, and is further characterised in that the coefficients corresponding to the base layer having a bandwidth limit are quantised and coded until a bit allocation is reached, and then the coefficients corresponding to an enhancement layer having a bandwidth limit are quantised and coded until a bit allocation is reached, the quantisation and coding being repeated until all layers have been coded.

30 According to another aspect of the present invention, there is provided a method for decoding a datastream representing an audio signal, comprising the steps of:

(a) initialising entries in a coefficient list to zero, where the list order preserves the frequency order of coefficients and groups together coefficients with the same frequency index;

5 (b) decoding bitplane data from the datastream in order of significance beginning with the most significant bitplane, where bitplane data corresponds to quantised coefficient bits of equal significance, and decoding of one or more bitplanes comprises the steps of:

10 (i) decoding runlength codes to locate newly-significant coefficient list entries which have most-significant magnitude bit (MSB) positions within the current bitplane;

15 (ii) setting magnitudes of said newly-significant coefficient list entries to a predetermined threshold level corresponding to the current bitplane;

(iii) decoding the signs of said newly-significant coefficient list entries;

20 (iv) removing said newly-significant entries from the coefficient list.

(c) reordering significant coefficients removed from the coefficient list to a set of frequency-domain output coefficients.

25 According to a further aspect of the present invention, there is provided a method for encoding audio signals to a layered datastream having a base layer and a predetermined number of enhancement layers, comprising the steps of:

30 (a) reordering frequency-domain coefficients representing an audio signal to a coefficient list, where the list order preserves the frequency order of

coefficients and groups together coefficients with the same frequency index;

5 (b) quantising and coding coefficients corresponding to the base layer with a predetermined bandwidth limit, until a predetermined bit allocation for the base layer is reached;

10 (c) quantising and coding coefficients corresponding to the next enhancement layer with a predetermined bandwidth limit, until a predetermined bit allocation for the enhancement layer is reached;

(d) sequentially performing step (c) until all layers have been coded, wherein steps (b), (c) and (d) each includes coding quantised coefficient bits of equal significance together in bitplanes, where bitplanes are coded in  
15 order of significance beginning with the most significant bitplane, and coding of one or more bitplanes comprises the steps of:

20 (i) locating newly-significant coefficients with most-significant magnitude bit (MSB) positions within the current bitplane, by runlength coding positions of coefficient list entries whose magnitudes equal or exceed a predetermined threshold level corresponding to the current bitplane;

(ii) coding the signs of said newly-significant coefficients;

25 (iii) removing said newly-significant coefficients from the coefficient list.

(e) outputting coded layer data to the datastream.

30 According to a further aspect of the present invention, there is provided a method for decoding audio signals from a layered datastream having a base layer and a predetermined number of enhancement layers, comprising the steps of:

- (a) initialising entries in a coefficient list to zero, where the list order preserves the frequency order of coefficients and groups together coefficients with the same frequency index;
- 5 (b) decoding data from the datastream corresponding to the base layer with a predetermined bandwidth limit, until a predetermined bit allocation for the base layer is reached;
- 10 (c) decoding data from the datastream corresponding to the next enhancement layer with a predetermined bandwidth limit, until a predetermined bit allocation for the enhancement layer is reached;
- 15 (d) sequentially performing step (c) until all layers have been decoded, wherein steps (b), (c) and (d) each includes decoding bitplane data corresponding to quantised coefficient bits of equal significance, where bitplanes are decoded in order of significance beginning with the most-significant bitplane, and decoding of one or more bitplanes comprises the steps of:
- 20 (i) decoding runlength codes to locate newly-significant coefficient list entries which have most-significant magnitude bit (MSB) positions within the current bitplane;
- 25 (ii) setting said newly-significant coefficient list entries to a predetermined threshold level corresponding to the current bitplane;
- (iii) decoding the signs of said newly-significant coefficient list entries;
- 30 (iv) removing said newly-significant entries from the coefficient list.

(e) reordering significant coefficients removed from the coefficient list to a set of frequency-domain output coefficients.

According to a further aspect of the present invention, there is provided a method for decoding audio signals from a layered datastream having a base layer and a predetermined number of enhancement layers, where decoding of each frame of coded data comprises the steps of:

(a) decoding data from the datastream and reconstructing output coefficients corresponding to the base layer with a predetermined bandwidth limit, until a predetermined bit allocation for the base layer is reached or all of the data for the frame has been decoded;

(b) decoding data from the datastream and reconstructing output coefficients corresponding to the next enhancement layer with a predetermined bandwidth limit, until a predetermined bit allocation for the enhancement layer is reached or all of the data for the frame has been decoded;

(c) sequentially performing step (b) until all layers have been decoded, or until all of the data for the frame has been decoded;

(d) transforming reconstructed output coefficients to a time-domain output signal;

(e) lowpass filtering the time-domain output signal, where the lowpass filter cutoff frequency is dependent on the bandwidth limit of the last layer decoded.

According to a further aspect of the present invention, there is provided an apparatus for encoding audio signals to a datastream, the apparatus comprising:

(a) reordering means for reordering frequency-domain coefficients representing an audio signal to a coefficient list, where the reordering means is configured to preserve the frequency order of coefficients within the list, and to grouping together coefficients with the same frequency index;

(b) bitplane coding means for quantising the coefficients and coding bits of equal significance together in bitplanes, where the bitplane coding means is configured to code bitplanes in order of significance beginning with the most-significant bitplane, and coding of one or more bitplanes comprises the steps of;

(i) locating newly-significant coefficients with most-significant magnitude bit (MSB) positions within the current bitplane, by runlength coding positions of coefficient list entries whose magnitudes equal or exceed a predetermined threshold level corresponding to the current bitplane;

(ii) coding the signs of said newly-significant coefficients;

(iii) removing said newly-significant coefficients from the coefficient list.

(d) means for outputting coded bitplane data to the datastream.

According to a further aspect of the present invention, there is provided an apparatus for decoding a datastream representing an audio signal, the apparatus comprising:

(a) means for initialising entries in a coefficient list to zero, where the list order preserves the frequency order of coefficients and groups together coefficients with the same frequency index;

(b) bitplane decoding means for decoding bitplane data from the datastream in order of significance beginning with the most significant bitplane, where bitplane data corresponds to quantised coefficient bits of equal significance, and decoding of one or more bitplanes comprises the steps of:

5

(i) decoding runlength codes to locate newly-significant coefficient list entries which have most-significant magnitude bit (MSB) positions within the current bitplane;

10

(ii) setting magnitudes of said newly-significant coefficient list entries to a predetermined threshold level corresponding to the current bitplane;

15

(iii) decoding the signs of said newly-significant coefficient list entries;

(iv) removing said newly-significant entries from the coefficient list.

(c) means for reordering significant coefficients removed from the coefficient list to a set of frequency-domain output coefficients.

20

According to a further aspect of the present invention, there is provided an apparatus for encoding audio signals to a layered datastream having a base layer and a predetermined number of enhancement layers, the apparatus comprising:

25

(a) means for reordering frequency-domain coefficients representing an audio signal to a coefficient list, where the list order preserves the frequency order of coefficients and groups together coefficients with the same frequency index;

30

(b) means for quantising and coding coefficients corresponding to the base layer with a predetermined bandwidth limit, until a predetermined bit allocation for the base layer is reached;

5 (c) means for quantising and coding coefficients corresponding to the next enhancement layer with a predetermined bandwidth limit, until a predetermined bit allocation for the enhancement layer is reached;

(d) means for sequentially performing step (c) until all layers have been  
10 coded, wherein steps (b), (c) and (d) each includes bitplane coding means for coding quantised coefficient bits of equal significance together in bitplanes, where the bitplane coding means is configured to code bitplanes in order of significance beginning with the most significant bitplane, and coding of one or more bitplanes comprises the steps of:

15

(i) locating newly-significant coefficients with most-significant magnitude bit (MSB) positions within the current bitplane, by runlength coding positions of coefficient list entries whose magnitudes equal or exceed a predetermined threshold level  
20 corresponding to the current bitplane;

20

(ii) coding the signs of said newly-significant coefficients;

25

(iii) removing said newly-significant coefficients from the coefficient list.

(f) means for outputting coded layer data to the datastream.

According to a further aspect of the present invention, there is provided an  
30 apparatus for decoding audio signals from a layered datastream having a base layer and a predetermined number of enhancement layers, the apparatus comprising:

(a) means for initialising entries in a coefficient list to zero, where the list order preserves the frequency order of coefficients and groups together coefficients with the same frequency index;

5 (b) means for decoding data from the datastream corresponding to the base layer with a predetermined bandwidth limit, until a predetermined bit allocation for the base layer is reached;

10 (c) means for decoding data from the datastream corresponding to the next enhancement layer with a predetermined bandwidth limit, until a predetermined bit allocation for the enhancement layer is reached;

(d) means for sequentially performing step (c) until all layers have been decoded, wherein steps (b), (c) and (d) each includes bitplane decoding  
15 means for decoding bitplane data corresponding to quantised coefficient bits of equal significance, where bitplanes are decoded in order of significance beginning with the most-significant bitplane, and decoding of one or more bitplanes comprises the steps of:

20 (i) decoding runlength codes to locate newly-significant coefficient list entries which have most-significant magnitude bit (MSB) positions within the current bitplane;

25 (ii) setting said newly-significant coefficient list entries to a predetermined threshold level corresponding to the current bitplane;

(iii) decoding the signs of said newly-significant coefficient list entries;

30 (iv) removing said newly-significant entries from the coefficient list.

(e) means for reordering significant coefficients removed from the coefficient list to a set of frequency-domain output coefficients.

According to a further aspect of the present invention, there is provided an apparatus for decoding audio signals from a layered datastream having a base layer and a predetermined number of enhancement layers, the  
5 apparatus for decoding each frame of coded data comprising:

(a) means for decoding data from the datastream and reconstructing output coefficients corresponding to the base layer with a predetermined bandwidth limit, until a predetermined bit allocation for the base layer is  
10 reached or all of the data for the frame has been decoded;

(b) means for decoding data from the datastream and reconstructing output coefficients corresponding to the next enhancement layer with a predetermined bandwidth limit, until a predetermined bit allocation for the  
15 enhancement layer is reached or all of the data for the frame has been decoded;

(c) means for sequentially performing step (b) until all layers have been decoded, or until all of the data for the frame has been decoded;

20 (d) means for transforming reconstructed output coefficients to a time-domain output signal;

(e) filter means for lowpass filtering the time-domain output signal, where  
25 the filter means is configured so that the lowpass filter cutoff frequency is dependent on the bandwidth limit of the last layer decoded.

The herein described methods allow the encoding of audio signals to a datastream with fine-grain bitrate scalability. The method involves  
30 reordering frequency-domain transform coefficients, and coding coefficient bitplanes in order of significance. Bitplane coding includes the steps of significance map coding and a refinement stage. Significance map coding identifies those coefficients with an MSB within the current bitplane by

arranging reordered coefficients into lists and runlength coding the positions of list entries that are newly significant at the current bitplane level. The refinement stage codes lower-significance bits of coefficients identified in earlier bitplanes.

5

Further, an apparatus encodes time-domain audio signals to a datastream with fine-grain bitrate scalability, the apparatus having means for transforming a time-domain signal to the frequency domain, weighting and reordering the transform coefficients, and coding coefficient bitplanes in  
10 order of significance. Means for bitplane coding includes the steps of significance map coding and a refinement stage. Means for significance map coding identifies those coefficients with an MSB within the current bitplane by arranging reordered coefficients into lists and runlength coding the positions of list entries that are newly significant at the current bitplane  
15 level. The means for refinement codes lower-significance bits of coefficients identified in earlier bitplanes.

In a method for decoding audio signals from a datastream, involving the steps of decoding data for each coded bitplane, and reordering reconstructed  
20 frequency-domain coefficients, bitplane data is decoded with knowledge of the algorithm used to code significance maps in the encoder. Because the encoded signal has been coded in bitplane order, the decoder can operate on any truncated code with a bitrate less than the encoded rate to provide a lower-quality output signal.

25

A decoding apparatus comprising means for decoding data for each coded bitplane, reordering and inverse weighting reconstructed coefficients, and inverse transforming coefficients to a time-domain output signal, operates with knowledge of the algorithm used to code significance maps in an  
30 encoder. Because the encoded signal has been coded in bitplane order, the decoding apparatus can operate on any truncated code with a bitrate less than the encoded rate to provide a lower-quality output signal.

Two classes of bitplane coding algorithm are considered. Fixed-bandwidth algorithms code a fixed bandwidth range of transform coefficients for all bitplanes, which results in datastreams where coding bandwidth is essentially invariant with decoded bitrate. Alternatively layered algorithms  
5 restrict the range of coefficient frequencies coded in bitplanes within lower-bitrate layers, and code higher-frequency information in higher layers. Layered bitplane coding results in increased coding bandwidth as decoded bitrate increases, and can result in improved subjective quality at lower bitrates.

10

In a first fixed-bandwidth bitplane encoding method, frames of quantised transform coefficients representing the input signal are each arranged in sign-magnitude format and reordered to a list of insignificant coefficients (LIC), where reordering clusters together coefficients with the same  
15 frequency index. The coefficients are then scanned in bitplane order beginning with the most-significant bitplane, and the positions of newly significant coefficients within the LIC identified by runlength coding for each bitplane. A sign bit is output following the runlength code for each new significant coefficient location, and the coefficient is moved from the  
20 LIC to a list of significant coefficients (LSC). Following completion of the LIC scan, LSC entries identified in earlier (more significant) bitplanes are refined for the current bitplane level.

A first fixed-bandwidth bitplane decoding method mirrors the operation of  
25 the encoding method. At the start of decoding each frame of data from a datastream, entries in a list of insignificant coefficients are reset to zero. Data is then decoded for each bitplane beginning with the most significant bitplane, and the positions of newly-significant LIC entries identified by decoding runlength codes for each bitplane. A sign bit is also decoded for  
30 each significant LIC entry, and the coefficients moved to a LSC. Refinement data is decoded to refine LSC entries identified in earlier bitplanes. Finally the reconstructed coefficients are reordered, inverse weighted and transformed to a time-domain output signal.

A second fixed-bandwidth bitplane encoding method follows the first encoding method but in addition within each bitplane scan extracts coefficients from the LIC which have a higher probability of becoming significant, to form a subsequence which is coded before coefficients that remain in the LIC. A new subsequence is conveniently formed at the beginning of each bitplane scan. Coefficient contexts used to form the subsequence include the presence of significant neighbour coefficients. As for LIC coding, subsequence coding is also performed using runlength codes. Coding the subsequence before the LIC for each bitplane improves coding efficiency for those frames where coding of the final bitplane is only partially completed. A second fixed-bandwidth bitplane decoding method mirrors the operation of the encoding algorithm.

Another method encodes audio signals in a layered manner, where a number of bitrate ranges are defined wherein bitplane scans are constrained to a limited range of coefficient frequencies. This results in a layered datastream where coding bandwidth increases with bitrate, and fine-grain scalability is maintained within each coded layer. The method involves transforming a time-domain signal to the frequency domain, weighting and reordering the transform coefficients, and layered bitplane encoding. Following coding of the base layer with the lowest bandwidth, coding of each enhancement layer includes coefficients to a new bandwidth limit and also codes uncoded data contained within previous layer bandwidth limits. Coding of each bitplane contained within a layer follows the approach established for fixed-bandwidth coding, including significance map coding and a refinement stage.

Layered datastreams may be decoded where coefficients are reconstructed to a progressively higher bandwidth as decoded bitrate increases. The method involves layered bitplane decoding, and subjecting reconstructed coefficients to inverse reordering and weighting processes before inverse transformation to a time-domain output signal. At lower decoded bitrates

where the final encoded layer is not decoded, the time-domain output signal is lowpass filtered to attenuate nonlinear artifacts caused by only partially decoding the full bandwidth range of encoded transform coefficients.

- 5 A first layered bitplane encoding method broadly follows the first fixed-bandwidth bitplane encoding method, except that the bandwidth of each bitplane scan is constrained to the bandwidth limit of the current layer. Quantised transform coefficients representing the entire bandwidth of the input signal are arranged in sign-magnitude format and reordered to a list of
- 10 insignificant coefficients (LIC), where reordering clusters together coefficients with the same frequency index. Each layer is then coded in bitplane order beginning with the most-significant bitplane, where each bitplane coding includes scans of both the LIC and a list of significant coefficients (LSC), and the number of LIC entries scanned depends on the
- 15 bandwidth limit for the current layer. For each bitplane, positions of newly-significant coefficients within the LIC are identified by runlength codes, followed by a sign bit for each new significant coefficient location. Significant coefficients are moved from the LIC to the LSC. Following completion of the LIC scan, LSC entries identified in earlier (more
- 20 significant) bitplanes are refined for the current bitplane level. Coding of the base layer with the lowest bandwidth is followed by enhancement layers with progressive increases in coding bandwidth, where each enhancement layer contains coded bitplane information to the new bandwidth limit and also uncoded data from earlier layers. A first layered bitplane decoding
- 25 method mirrors the operation of the encoding algorithm.

- A second layered bitplane encoding method may follow the procedure of the first layered bitplane encoding method but in addition within each bitplane scan forms a subsequence of coefficients extracted from the LIC,
- 30 which is coded before those coefficients that remain in the LIC. A new subsequence is conveniently formed at the beginning of each bitplane scan within each layer. A second layered bitplane decoding method mirrors the operation of the encoding algorithm.

Methods are described for efficiently coding audio transform coefficient bitplanes. The methods achieve high coding efficiency such that audio signals are compressed to relatively compact representations. The coding  
5 methods can be executed with algorithms that offer low computational complexity, and do not require Huffman or arithmetic coding.

It will be realised that both the coding and decoding apparatuses described herein may be constituted using a variety of computation means, including  
10 distributed systems, well known to those skilled in the art.

### **Brief Description of the Drawings**

The invention will now be described, by way of examples which are not  
15 intended to be limiting, and with reference to the accompanying drawings, of which;

Figure 1 is a block diagram showing an audio encoding apparatus that uses bitplane encoding, according to the first embodiment of the invention.  
20

Figure 2 illustrates an example frequency-domain transform output corresponding to one frame of audio input data for an encoder using a block-switched modified discrete cosine transform.

25 Figure 3 shows an example nonuniform time-frequency decomposition of a wavelet packet transform for use in an audio encoder.

Figure 4 is a flowchart illustrating the operation of a general bitplane encoding algorithm for use in audio encoding apparatus according to the  
30 first embodiment of the invention.

Figure 5 illustrates the significance map and refinement elements of a bitplane coding process.

Figure 6 is a block diagram showing an audio decoding apparatus that uses bitplane decoding, according to the first embodiment of the invention.

5 Figure 7 is a flowchart illustrating the operation of a general bitplane decoding algorithm for use in audio decoding apparatus according to the first embodiment of the invention.

10 Figure 8 is a flowchart illustrating the operation of a fixed-bandwidth bitplane encoding algorithm for use in audio encoding apparatus according to the second embodiment of the invention.

15 Figure 9 is a flowchart illustrating the operation of the significance map encoding stage of a fixed-bandwidth encoder according to the second embodiment of the invention.

Figure 10 is a flowchart illustrating the operation of a fixed-bandwidth bitplane decoding algorithm for use in audio decoding apparatus according to the second embodiment of the invention.

20 Figure 11 is a flowchart illustrating the operation of the significance map decoding stage of a fixed-bandwidth decoder according to the second embodiment of the invention.

25 Figure 12 is a flowchart illustrating the operation of a fixed-bandwidth bitplane encoding algorithm for use in audio encoding apparatus according to the third embodiment of the invention.

30 Figure 13 illustrates the coding order for a layered bitplane encoding process in the frequency domain, where each new layer codes coefficients to a new bandwidth limit and also codes uncoded data within previous layer bandwidth limits.

Figure 14 is a block diagram showing an audio encoding apparatus using layered bitplane encoding, according to the fourth embodiment of the invention.

- 5 Figure 15 is a block diagram showing an audio decoding apparatus using layered bitplane decoding, and including a lowpass output filter, according to the fourth embodiment of the invention.

- 10 Figure 16 illustrates nonlinear high-frequency artefacts at the output of a layered bitplane decoder when data for the final encoded layer is not decoded.

- 15 Figure 17 is a flowchart illustrating the operation of a layered bitplane encoding algorithm for use in audio encoding apparatus according to the fifth embodiment of the invention.

- 20 Figure 18 is a flowchart illustrating the operation of a layered bitplane encoding algorithm for use in audio encoding apparatus according to the sixth embodiment of the invention.

Figure 19 is a block diagram showing an audio transcoding apparatus with a datastream input and a scalable datastream output, according to the seventh embodiment of the invention.

- 25 Figure 20 is a block diagram showing an audio transcoding apparatus with a scalable datastream input and a datastream output, according to the seventh embodiment of the invention.

### **Detailed Description of Preferred Embodiments**

30

Referring to figure 1, an encoding apparatus comprises an audio input unit 101, a time-frequency transform unit 102, a scaling and weighting unit 103,

a psychoacoustic model unit 104, a bitplane encoding unit 105, and a datastream output unit 106.

5 In the description of this embodiment it is assumed that single-channel (monaural) sampled (discrete-time) audio data having 16 signed integer bits per sample is to be encoded. It is further assumed that the sampling rate of the audio data is sufficient to support the full audio spectrum of 0 to 20 kHz, for example a sampling rate of 48 kHz. However, the invention is not limited thereto, but is also applicable to encoding single-channel audio data  
10 with other resolutions and sampling rates, for example 12-bit data sampled at 16 kHz. The invention is also applicable to encoding multi-channel audio data.

15 The operation of each unit of the embodiment will be described in detail.

Audio data to be encoded is successively input from the audio input unit 101 as frames of time-domain samples. The audio input unit 101 may be the output interface of an analog-to-digital converter (ADC) used to digitise a continuous-time (analog) audio signal, an interface to a hardware network,  
20 or the like. The audio input unit 101 may also be a storage device such as a RAM, a ROM, a hard disk, and a CD-ROM. A typical frame length is 1024 samples.

Time-domain data from the audio input unit 101 is converted to frequency-domain data by the time-frequency transform unit 102. One possible form of transform is the modified-discrete cosine transform (MDCT) (as used in MPEG-2 AAC for example), where adjacent blocks of samples are windowed and transformed to the frequency domain. For a frame of  $K$  time-domain input samples, the frequency-domain transform output can be  
25 arranged as a series of  $B$  blocks, each with  $M$  coefficients ranging from dc to half the sampling frequency, where to preserve critical sampling  $K = BM$ .  
30

MDCT transform coefficients can be indexed with a frequency index  $m$ , and time index  $b$ :

$$\text{MDCToutput} = X[m][b],$$

$$\text{where } m = 0 \dots M - 1$$

$$b = 0 \dots B - 1.$$

- $B$  and  $M$  respectively determine the time and frequency resolution of the transform output in each frame – higher  $B$  results in better time resolution, whereas increasing  $M$  improves frequency resolution. Time / frequency resolution can be adapted to the characteristics of the input signal by using block switching, where Fig. 2 shows longer transform windows 201 (larger  $M$ ) used for stationary signal frames, and shorter window lengths 202 (smaller  $M$ ) used under transient conditions.

- An alternative to the block-switched MDCT is the wavelet packet (WP) transform, which can be arranged to achieve a nonuniform decomposition where time and frequency resolution vary as a function of frequency.
- Increasing the time resolution at the expense of frequency resolution for higher-frequency subbands can achieve a time-frequency resolution that approximates that of the hearing system, allowing good transient performance without the use of block switching.
- $M$ -band wavelet packet transform coefficients can be indexed with a subband frequency index  $m$ , and time index  $b$ , where the number of subband samples per frame  $B_m$  depends on the decomposition depth for each subband:

$$\text{WPoutput} = X[m][b],$$

$$\text{where } m = 0 \dots M - 1$$

$$b = 0 \dots B_m - 1.$$

For critical sampling the following relationship holds for the WP transform:

$$\sum_{m=0}^{M-1} B_m = K .$$

Fig. 3 shows an example 29-band wavelet packet decomposition for use in an encoder, where each of the tree branches represents a lowpass-highpass filter pair and decimation process. If this transform is used with a frame length of 1024 samples, the lowest- frequency subband outputs will contain 4 samples per frame ( $B_m = 4$ ), while the highest frequency subband outputs will contain 128 samples per frame ( $B_m = 128$ ).

It is also possible to obtain a nonuniform decomposition with an MDCT-based system by combining high-frequency coefficients.

In a scalable compression system it is desirable to quantise and code the transform output for each frame in an embedded manner, allowing the resultant datastream to be truncated to a lower-rate representation that remains decodable. Embedded coding is conveniently achieved using bitplane coding. One of the characteristics of bitplane coding is that because in each bitplane scan the same threshold level is used to construct codes for all coefficients, the resultant quantisation error will tend to a white spectrum. Such an error characteristic is sub-optimal for audio coding because masking results in a nonuniform spectral sensitivity to quantisation error. Spectral error shaping can reduce error audibility, and can be achieved by weighting the transform output prior to bitplane encoding, and performing an inverse weighting at the decoder following bitplane decoding.

Referring again to Fig. 1, in the present embodiment the transform output coefficients are input to a scaling and spectral weighting unit 103 prior to bitplane encoding. In general the transform output coefficients will be in floating-point format even if the time-domain input samples are of integer

format. A scaling operation scales the magnitudes of all transform coefficients in a frequency-independent manner so that they occupy a sufficiently large integer range prior to bitplane encoding. The scaling operation is fixed and does not change from frame to frame. A spectral  
5 weighting operation provides a frequency-dependent weighting of scaled transform coefficients  $X(k)$ ,

$$X'(k) = \frac{X(k)}{W(k)}, \quad \text{for } k = 0 \dots K - 1.$$

where the weighting function  $W(k)$  follows the desired error shaping function, and  $X'(k)$  represents the scaled and weighted transform  
10 coefficients. One approach is to set the weighting function for each frame so that error shaping approximates the masked threshold for the frame, determined by a psychoacoustic model unit 104. The weighting function is coded to the datastream as side information for each frame so that a decoder can provide the correct inverse weighting. The overhead corresponding to  
15 weight side information can be minimised by quantising and entropy coding the weighting function across banded coefficient groups. An example weighting scheme consists of 32 band weights quantised in 3.0 dB steps, where the band widths approximate the critical band law of the hearing process.

20 The scaled and weighted transform coefficients  $X'(k)$  are then input to a bitplane encoding unit 105, where coefficient bits of equal significance are grouped together into bitplanes, and each bitplane coded in order of significance.

25 A general bitplane-encoding algorithm 105 is shown as a flowchart in Fig. 4. At step s401 a bit allocation variable is initialised to the required size of the coded frame, and is subsequently updated as data is coded in order to indicate the number of bits which are available to code further data for the  
30 frame. Scaled and weighted floating-point transform coefficients  $X'(k)$  are represented in sign-magnitude format, and at step s402 the largest

coefficient magnitude within the frame  $|X'|_{\max}$  is determined and an initial threshold level  $T$  set such that

$$T \leq |X'|_{\max} < 2T.$$

5  $T$  determines the current bitplane level in the encoding process, and the most significant bitplane within the frame is coded with the initial threshold value. The initial threshold value is output as side information to an output buffer for coded frame data, so that a decoder receiving a coded datastream can begin decoding at the correct bitplane level.

10 For each bitplane coefficients are scanned at step s403 to locate those with magnitudes equal to or exceeding  $T$  – these coefficients are termed ‘significant’ with respect to the current threshold. With reference to Fig. 5, data describing newly significant coefficient locations within each bitplane – ie the positions of coefficients that have their MSB located within the current bitplane is termed a ‘significance map’. When a significant  
15 coefficient is located, the component of the significance map describing the location is coded and output to the output buffer, followed by a sign bit representing the sign of the coefficient. Less-significant bits of significant coefficients are termed ‘refinement’ bits.

20 When all of the transform coefficients have been scanned at the initial threshold level,  $T$  is halved at step s405 and coding progresses to the next bitplane where all coefficients not yet found to be significant are scanned using the new value of  $T$ . For each new significant coefficient identified a significance map component and sign bit are coded and output to the output  
25 buffer at step s403. When this second significance map is complete a refinement stage s404 is executed where refinement bits corresponding to the new threshold level are output to the output buffer for all significant coefficients identified in the first bitplane scan. The threshold is halved again, and significance map and refinement data coded for the third  
30 bitplane. This process is repeated for progressively less significant bitplanes until the bit allocation for the frame is reached, at which point coding

terminates (step s406), and at step s407 coded frame data in the output buffer is written to the datastream output unit 106.

In effect the general bitplane coding algorithm described implements  
5 uniform quantisation with a dead-zone around zero, where integer quantised coefficient values are given by

$$q(k) = \text{sgn}(X'(k)) \left\lfloor \frac{X'(k)}{T_F} \right\rfloor,$$

and  $T_F$  is the final threshold value used to code each coefficient.

10 In general, significance map coding at step s403 is achieved in embodiments of the present invention by forming lists of coefficients, testing list entries for significance with respect to threshold  $T$ , and outputting significance test results to the output buffer. A simple coding approach is to output a single bit for each list entry tested – for example, ‘0’  
15 and ‘1’ could indicate insignificant and significant entries respectively. However, unless the probability of significance  $s$  is close to 0.5 then this coding method is relatively inefficient. Often  $s \ll 0.5$ , in which case improved coding efficiency can be achieved by runlength coding the significant entry locations within a list.

20

A useful runlength code is the Golomb code with parameter  $p$ , where non-negative runlength  $r$  is coded as 2 components – a prefix  $\lfloor r / p \rfloor$  coded in unary, followed by suffix  $[r \bmod p]$  coded in binary. A particularly simple form of Golomb code, sometimes known as Rice codes, occurs when  $p = 2^n$   
25 for some integer  $n \geq 0$  - here  $r$  can be coded by removing the  $n$  least-significant bits from  $r$ , coding the remainder as a unary prefix, and appending  $n$  binary LSB's. For example, if  $r = 9$  and  $n = 2$ , then the Golomb-Rice code for  $r$  is ‘00101’ – here the prefix is ‘001’ = 8, and the remainder is ‘01’ = 1.

30

It should be noted that the configuration of Golomb runlength codes is not limited only to that used in the above embodiment, where the variable-length prefix is coded as '0's followed by a '1', and is followed by the fixed-length suffix. Instead, the use of '0's and '1's may be reversed to code  
5 the variable-length part. Further, the Golomb code may be coded as a fixed-length part followed by a variable-length part.

The coding efficiency achieved using Golomb-Rice codes to runlength code significant entry locations in a list depends on the code wordlength  $n$  and  
10 the runlength distribution.  $n$  can be set to a fixed value which on average results in the most compact list code across many frames of a test item. Alternatively  $n$  can be optimised for each frame, and sent as side information at the start of the frame so that a decoder can correctly interpret the coded list data. Yet another approach is to optimise  $n$  for each bitplane  
15 of each frame, and send the appropriate side information at the start of each bitplane.

A different approach to adapting the runlength coder wordlength to the runlength statistics of a list is to make the Golomb-Rice code adaptive in  
20 the sense that  $n$  varies as a function of list data coded – that is, backwards-adaptive runlength coding. An adaptive code such as that described by Langdon Jr could be used ("An Adaptive Run-Length Coding Algorithm," IBM Technical Disclosure Bulletin, vol. 26, pp. 3783 - 3785 (1983 Dec.)), where each '0' in the unary-coded prefix causes the wordlength  $n$  to  
25 increment, and  $n$  is decremented following the binary-coded suffix. For example, consider the code for  $r = 9$  with an initial wordlength  $n = 2$ : the adaptive Golomb-Rice code for  $r$  is 01101 – here the prefix is '01' = 4, the 3-bit remainder is '101' = 5, and the final value for  $n$  is 2. While this  
30 example considers the case where  $n$  increases or decreases by 1 for each prefix '0' or suffix output, it is also possible to construct adaptive runlength codes which adapt at different rates to adaptation instances – for example, where  $n$  increases by 1 for every second prefix '0' output (as described in WO0059116, Malvar). Of course other adaptation strategies also exist (E.

Ordentlich, M. Weinberger, and G. Seroussi, "A Low-Complexity Modeling Approach for Embedded Coding of Wavelet Coefficients," Proc. IEEE 1998 Data Compression Conference, Snowbird, Utah, pp. 408 – 417 (1998 Mar.)). The advantages of using adaptive Golomb-Rice codes to scan  
5 for significant entries within lists includes the simplicity and computational efficiency of the codes, and also the efficiency with which the codes can adapt to changing runlength statistics within a list, which results in relatively compact list coding without the use of wordlength side information.

10

Another form of adaptive runlength code is the exponential-Golomb code, or exp-Golomb code (J. Teuhola, "A Compression Method for Clustered Bit-Vectors," Information Processing Letters, vol. 7, pp. 308 – 311 (1978 Oct.)). Here the code wordlength  $n$  is set to a fixed value at the start of each  
15 code, and increments for each prefix '0' coded. An interesting aspect of exp-Golomb codes is that with minor modifications they can form reversible variable length codes (RVLCs), where the code prefix can be decoded in either a forward or reverse direction (J. Wen and J. D. Villasenor, "Reversible Variable Length Codes for Efficient and Robust  
20 Image and Video Coding," Proc. 1998 IEEE Data Compression Conference, pp. 471 – 480, Snowbird, Utah (1998 Mar.)). RVLCs can improve coding robustness with error-prone transmission channels. Note that RVLCs with the same length distributions as fixed-wordlength Golomb-Rice codes can also be formed.

25

When fixed- or adaptive- Golomb-Rice codes are used to scan lists for significant entries, coding the end of the list scan following the final significant entry location can be simply achieved by outputting a series of prefix '0's until the end of the list is passed. When a decoder receives coded  
30 list data and the current list position passes the known list length, all remaining list entries following the last significant position are marked as insignificant and decoding of the current list terminates. End-of-run codes in this fashion are particularly compact when an adaptive Golomb-Rice

code is used. For example, with the runlength adaptation rule described above and a list length of 1024 entries, end-of-run codes are represented with a maximum of eleven prefix '0's.

- 5    Returning again to Fig. 1, the final stage of the encoding process is to transmit to a memory or an external apparatus the bitplane encoded data stored in the output buffer, and also side information such as banded weight data, by means of a datastream output unit 106. The datastream output unit 106 may be a storage device such as a hard disk, a RAM, and a CD-ROM,  
10   or an interface to a public telephone line, a radio line, a LAN or the like.

Fig. 6 is a block diagram of an audio decoding apparatus also according to the first embodiment of the invention. The decoding apparatus comprises a datastream input unit 601, a bitplane decoding unit 602, an inverse scaling and weighting unit 603, a frequency-time transform unit 604, and an audio  
15   output unit 605.

Coded data frames representing bitplane-encoded audio data are received by a datastream input unit 601. The datastream input unit 601 may be a  
20   storage device such as a hard disk, a RAM, and a CD-ROM, or an interface to a public telephone line, a radio line, a LAN or the like.

The coded data is input to a bitplane decoding unit 602 that reconstructs transform coefficients in bitplane order. Fig. 7 shows a general bitplane  
25   decoding algorithm 602, where decoding of each frame begins at step s701 by storing coded data for the frame in an input buffer, and using the amount of coded data read for the frame to initialise a bit allocation variable. Before decoding the first bitplane the output coefficient values are reset to zero at step s702. An initial threshold level  $T$  read from the input buffer at step  
30   s703 then determines the most significant (initial) bitplane level for the frame. For each bitplane scan, significance map data read and decoded from the input buffer at step s704 identifies coefficients whose MSB is located in the current bitplane and the corresponding sign bits. These coefficients are

set to  $\pm T$  as appropriate to the sign of the coefficient. Refinement data decoded at step s705 is used to refine lower-significance bits of significant coefficients identified in previous bitplanes, and for each refinement bit received  $\pm T$  is added to the decoded coefficient value as appropriate. When  
5 data for the current bitplane has been read and decoded,  $T$  is halved at step s706 and decoding progresses to the next bitplane. This process continues until no more coded data is available to decode, at which point decoding for the current frame terminates at step s707. Because the coded data has been encoded in bitplane order, the decoder can reconstruct data to a lower  
10 precision simply by discarding data for less-significant bitplanes.

For each significant coefficient identified in the decoding process there exists a range of uncertainty concerning the reconstructed value, which depends on the threshold  $T_F$  corresponding to the final significant bit  
15 decoded for the coefficient. A simple reconstruction approach is to set each significant coefficient to the center of its uncertainty interval at step s708 by adding  $\pm 0.5T_F$ , depending on the sign of the coefficient.

Referring again to Fig. 6, decoded bitplane data is input to an inverse  
20 scaling and weighting unit 603, where a fixed and frequency-independent scaling operation complementary to that implemented in the encoder is applied to all decoded coefficient values  $\tilde{X}'(k)$ . Side information received from the datastream input unit 601 and representing banded weight values  $W(k)$  is then used to implement an inverse weighting process on the scaled  
25 coefficients,

$$\tilde{X}(k) = \tilde{X}'(k)W(k), \quad \text{for } k = 0 \dots K-1,$$

where  $\tilde{X}(k)$  represents reconstructed coefficient values following inverse scaling and weighting.

A frequency-time transform unit 604 then transforms blocks of decoded coefficients  $\tilde{X}(k)$  to the time domain. If the transform is an inverse modified discrete cosine transform (IMDCT) this process involves transforming and windowing adjacent blocks of coefficients to the time domain. An alternative transform is the inverse wavelet packet transform.

Time-domain data representing decoded sampled (discrete-time) data for each frame is output using an audio output unit 605. The audio output unit 605 may be a digital-to-analog converter (DAC) used to convert decoded data to a continuous-time audio signal, an interface to a hardware network, or the like. The audio output unit 605 may also be a storage device such as a RAM, a hard disk, and a CD-ROM.

Referring once again to Fig. 1, the bitplane encoding unit 105 codes coefficient bitplanes in order of significance. The operation of an example bitplane encoding process 105 for one frame of audio data using a fixed-bandwidth bitplane coding algorithm of the second embodiment of the invention will now be described with reference to Fig. 8.

The first step s801 of the encoding algorithm initialises a bit allocation variable for the frame. Then at step s802 transform coefficients are reordered to a list of insignificant coefficients (LIC), and at step s803 a list of significant coefficients (LSC) is initialised to an empty list. Then for each bitplane, beginning with the most significant bitplane determined at s804, a runlength coder is used to identify newly-significant coefficient locations within the LIC (step s805), followed by a refinement stage s806 that outputs less significant bits of significant coefficients identified in earlier bitplanes.

The reordering step s802 involves mapping scaled and weighted transform coefficients  $X'(k)$  representing the frame to the LIC in a data-independent manner such that coefficients with the same frequency index are clustered

together within the LIC. Since coefficients with the same frequency index tend to be of similar magnitude, the reordering operation has the effect of clustering significant coefficient locations within LIC scans. This results in longer runs of insignificant coefficients which can improve coding efficiency when runlength coded at step s805, particularly when using adaptive runlength codes.

When the transform is an MDCT the reordering at s802 can be described by

$$LIC(k) = X'[m][b], \quad \text{for } k = 0 \dots K-1,$$

$$\text{where } m = \left\lfloor \frac{k}{B} \right\rfloor, \\ LIC(k) = X'[k][0].$$

$$b = k \bmod B.$$

- 10 When a frame contains only a single block of MDCT coefficients (long block mode,  $B = 1$ ), the reordering operation is the trivial task of copying the coefficients to the LIC in frequency order,

- 15 When a frame contains several short MDCT blocks ( $B > 1$ ), coefficients with the same frequency index are clustered (grouped) together within the LIC. This operation can be viewed as a short block interleaving process.

- 20 A similar mapping is made when the transform is a wavelet packet transform, grouping together all coefficients with the same subband frequency index within the LIC.

- 25 Note that the above embodiment describes the case where the full-bandwidth range of transform coefficients is mapped to the LIC, and the LIC length is equal to the frame length  $K$ . In this case coding of each bitplane will cover a full-bandwidth set of coefficients. However a reduced-bandwidth set of coefficients can also be coded by discarding high-frequency coefficients from each block in the reordering process, in which

case the LIC length will be less than  $K$ . For both cases the coding bandwidth is constant for all bitplanes within a frame.

- Following coefficient reordering at step s802 and LSC initialisation at step s803, the magnitude of the largest LIC entry is used to set an initial threshold level  $T$  at step s804, which determines the most significant bitplane. Before coding the first bitplane,  $T$  is output to an output buffer at step s804.
- 5
- 10 Significance map coding of each bitplane (s805) involves scanning the LIC for significant entries, using runlength codes to determine  $k$  for which  $LIC(k) \geq T$ . Fig. 9 shows a flowchart illustrating the significance map encoding step s805. At step s901 the current bitplane scan position is initialised to the start of the LIC by setting  $j = 0$ . At step s902 the remaining
- 15 LIC members are scanned for significant entries where  $|LIC(k)| \geq T$ , and  $k$  set to the first significant entry at or beyond  $j$ . If significant entries exist, the runlength between  $j$  and  $k$  is calculated, and the number of bits required to code the runlength and sign bit calculated at s903. If sufficient bits remain from the overall bit allocation for the frame (s904), the runlength code is
- 20 output to the output buffer (s905) followed by a coefficient sign bit (s906). The significant coefficient is then moved from the LIC to the list of significant coefficients (LSC) at step s907, and will be scanned during refinement passes in future (less significant) bitplanes. The current scan position is updated to point to the next LIC entry at step s908, and coding
- 25 progresses to the next significant LIC entry. During significance map scans of less-significant bitplanes, the LIC position previously occupied by the significant coefficient at  $k$  is skipped, and does not contribute to future runlength codes.
- 30 If the test at s904 indicates insufficient bits remain from the overall bit allocation with which to code the next runlength code and sign bit of the LIC scan, then zeros are output to the output buffer until the bit allocation for the frame is reached (s912), and coding for the current frame terminates.

The end of each LIC scan following the final significant LIC entry can be simply coded by outputting a runlength code which causes the bitplane scan position to pass the end of the LIC (s909, s910, s911). As discussed above  
5 in the first embodiment, when Golomb codes are used for runlength coding, the end of an LIC scan can be compactly coded by outputting a series of prefix '0's until the end of the LIC is passed.

The coding efficiency of the significance map stage is determined by the  
10 runlength statistics of each bitplane and the runlength coding used. If a fixed Golomb-Rice runlength code is used where wordlength  $n$  is fixed for all bitplanes of all frames, an optimal value for  $n$  is selected which on average results in the most compact significance map code. Alternatively  $n$   
15 can be optimised for each frame at a certain target bitrate and sent as side information at the start of each frame, or optimised for each bitplane of each frame and sent as side information at the start of each bitplane.

An alternative to using a fixed runlength code is to use an adaptive Golomb-Rice code where wordlength  $n$  varies as a function of data coded in  
20 each bitplane. A suitable adaptation strategy is to increment  $n$  for each runlength prefix '0' bit output, and to decrement  $n$  following a runlength suffix code. For bitplanes of audio transform coefficients the average spacing between significant LIC entries tends to increase with frequency, and coding efficiency is improved by resetting the runlength coder  
25 wordlength to a small value at the beginning of each bitplane scan (step s901) – in practice resetting  $n$  to 0 or 1 yields good results.

Referring again to Fig. 8, whereas the LIC scan at step s805 determines the MSB position of each significant coefficient, LSB information is provided  
30 during refinement stage scans of the LSC (s806). For each LSB coded the probability of coding a '1' is close to that of coding a '0', hence the LSC scan is efficiently performed by outputting a single bit for each list entry. If insufficient bits remain from the overall bit allocation for the frame with

which to code the next refinement bit, then zeros are output until the bit allocation for the frame is reached, and coding for the current frame terminates.

- 5 Following significance map (LIC) and refinement (LSC) scans at the current threshold level  $T$ , coding for the current bitplane is complete.  $T$  is then halved at step s807 and coding continues to the next bitplane. Coding terminates when, at any stage within a bitplane scan, the bit allocation for the frame is reached (for example, at step s808). When coding for the
- 10 current frame is completed, output buffer data is written to the datastream output unit 106 at step s809, and the buffer emptied in anticipation of the next frame to be coded.

Referring again to Fig. 6, the bitplane-decoding unit 602 decodes bitplane-coded coefficient data, where each bitplane contains information for a

15 fixed-bandwidth coefficient set. The operation of an example bitplane decoding process according to the second embodiment of the invention will now be described with reference to Fig. 10.

- 20 In general the decoding algorithm mirrors the operation of the encoding algorithm (Fig. 8). At step s1001 a frame of coded data is read from the datastream input unit 601 to an input buffer, and a bit allocation variable initialised to the amount of data read for the frame. At s1002 a  $K$ -sample LIC is initialised and member coefficients reset to zero. A LSC is initialised
- 25 to an empty list at step s1003. An initial threshold value  $T$  corresponding to the most significant bitplane for the frame is read from the input buffer at s1004, and coefficients reconstructed in bitplane order by decoding LIC significance map data (s1005) and LSC refinement data (s1006) for each bitplane. At the end of decoding each bitplane,  $T$  is halved (s1007) and
- 30 decoding progresses to the next bitplane. When no more bits are available in the input buffer to decode, bitplane decoding terminates (s1008), and significant coefficients are set within their uncertainty intervals at step s1009. Finally coefficients are reordered to an output coefficient set at

s1010, where reordering de-interleaves coefficients with the same frequency index to their respective transform blocks.

Fig. 11 shows a flowchart illustrating the significance map decoding stage at s1005. At step s1101 the scan position for the current bitplane is initialised to the start of the LIC by setting  $j = 0$ . Also at s1101 if fixed Golomb-Rice runlength codes are to be decoded, any wordlength side information for the runlength decoder is read from the input buffer as required. If adaptive Golomb-Rice codes are to be decoded, the decoder wordlength is reset to a small value at s1101. Then at s1102 a runlength code is read from the input buffer and decoded. If bits remain in the input buffer for the current frame (s1103), an LIC index  $k$  is obtained at s1104 by adding the runlength to  $j$ . If  $k$  is within the bounds of the LIC (s1105), a sign bit is read from the input buffer (s1106) and the MSB of the LIC member set to the current bitplane by setting  $LIC(k) = \pm T$  as appropriate (s1107). The significant coefficient is then moved from the LIC to the LSC at step s1108, the current scan position updated to the next LIC position (s1109), and the next runlength code read and decoded from the datastream at step s1102. Note that if the LIC index  $k$  is beyond the end of the LIC at s1105, then significance map decoding for the current bitplane terminates.

Referring again to Fig. 1, the bitplane encoding unit 105 codes a fixed-bandwidth set of coefficients in each bitplane. The operation of an example bitplane encoding process 105 using a fixed-bandwidth bitplane coding algorithm of the third embodiment will now be described. This bitplane encoding method is similar to that of the second embodiment, but is enhanced by extracting LIC coefficients to form a subsequence which for each bitplane significance scan is coded before the remaining LIC coefficients.

30

With reference to Fig. 12, the first step s1201 of the encoding algorithm initialises a bit allocation variable for the frame. Then at step s1202 transform coefficients are reordered to a list of insignificant coefficients

(LIC), where as before the reordering process groups together coefficients with the same frequency index within the LIC. At step s1203 a list of significant coefficients (LSC) is initialised to an empty list. Then, for each bitplane beginning with the most significant bitplane determined at s1204, a  
5 significance map is coded in three stages s1205, s1206 and s1207, which collectively identify coefficients with most-significant magnitude bits in the current bitplane. Also for each bitplane, a refinement stage s1208 outputs less significant bits of significant coefficients identified in earlier bitplanes. The threshold  $T$  corresponding to the current bitplane is halved at s1209,  
10 and coding progresses to the next bitplane. When the bit allocation has been used, coding for the current frame terminates (s1210), and at step s1211 the contents of the output buffer is written to the datastream output unit 106.

The criteria used to extract coefficients from the LIC to form a subsequence  
15 at s1205 is that the extracted coefficients should have a higher expected probability of becoming significant in the pending bitplane scan than those coefficients that remain in the LIC. Suitable contexts for selecting coefficients to form the subsequence include:

- coefficients that are frequency-domain neighbours to significant  
20 coefficients with the same time index
- for frames containing more than one transform block ( $B > 1$ ), coefficients that are time-domain neighbours to significant coefficients with the same frequency index
- the significant neighbour 'age', or the bitplane difference between the  
25 significant neighbour MSB bitplane and the current bitplane
- coefficients that have some harmonic relationship to significant coefficients.

Note that while coefficient extraction can in theory take place at any point(s) within a bitplane scan, in practice a convenient point at which to  
30 form the subsequence is at the start of each bitplane scan (as shown for s1205 in Fig. 12).

The subsequence is coded at s1206 by scanning for significant entries using runlength codes. A suitable runlength code is a fixed or adaptive Golomb code. When a significant entry is found a sign bit is output to the output buffer and the coefficient is moved from the subsequence to the LSC.

- 5 Coding the subsequence at s1206 before the remaining LIC coefficients at step s1207 results in improved coding efficiency for those frames where the final bitplane scan is only partially completed.

- 10 Following subsequence coding, the LIC is scanned for significant entries at s1207, also using runlength codes in a similar fashion to the method of the second embodiment (Fig. 9). If a fixed Golomb-Rice runlength code is used where wordlength  $n$  is fixed for all bitplanes of all frames, an optimal value for  $n$  is selected which on average results in the most compact significance map code. Alternatively  $n$  can be optimised for each frame at a certain  
15 target bitrate and sent as side information at the start of each frame, or optimised for each bitplane of each frame and sent as side information at the start of each bitplane. If an adaptive Golomb-Rice code is used then  $n$  is reset to a small value at the beginning of each bitplane scan (ie beginning of step s1207).

- 20 Referring once again to Fig. 6, the bitplane-decoding unit 602 decodes bitplane-coded coefficient data, where each bitplane contains information for a fixed-bandwidth coefficient set. The operation of a bitplane decoding process 602 according to the third embodiment of the invention (not shown)  
25 is similar to that of the second embodiment (Fig. 10), except that for each bitplane the significance map is decoded in three steps to mirror the operation of the encoding process shown in Fig. 12. Hence for each bitplane, significance map decoding comprises the steps of subsequence formation using the same context rules used in the encoder, decoding  
30 subsequence runlength codes and sign bits, and decoding LIC runlength codes and sign bits.

- The fixed-bandwidth coding algorithms described for previous embodiments code a fixed frequency range of transform coefficients together in each bitplane, where coding bandwidth is invariant with bitrate. While fixed-bandwidth coding results in good subjective quality at higher
- 5 bitrates, coding quality can decrease at lower bitrates where on average fewer bits are available to code each significant coefficient. At lower bitrates improved subjective quality can be achieved by limiting the bandwidth of each bitplane scan, essentially because on average more bits are allocated to each significant coefficient coded. Ideally the coding
- 10 bandwidth should be constrained to a fixed value within a defined bitrate range, so that consecutive frames decoded at the same bitrate have the same bandwidth. This avoids consecutive frames being decoded to different bandwidths, which can result in uncanceled transform alias products.
- 15 Defining a number of bitrate ranges where encoder bitplane scans are constrained to a limited range of coefficient frequencies results in a 'layered' datastream where coding bandwidth increases with bitrate, and fine-grain scalability is maintained within each coded layer. Referring to Fig. 13, following coding of the base layer with the lowest bandwidth, each
- 20 enhancement layer codes coefficients to a higher bandwidth limit, and can also code uncoded coefficient data from previous layer bandwidth limits.

Fig. 14 is a block diagram of an audio encoding apparatus according to the fourth embodiment of the invention. The encoding apparatus comprises an

25 audio input unit 1401, a time-frequency transform unit 1402, a scaling and weighting unit 1403, a psychoacoustic model unit 1404, a layered bitplane encoding unit 1405, and a datastream output unit 1406. These processes operate in a similar manner to those of the first embodiment (Fig. 1), except that bitplanes are encoded in a layered manner by the layered bitplane

30 encoding unit 1405, and the datastream output unit 1406 interleaves layered bitplane data with banded weight side information to yield a layered datastream output.

Fig. 15 is a block diagram of an audio decoding apparatus according to the fourth embodiment of the invention. The decoding apparatus comprises a datastream input unit 1501, a layered bitplane decoding unit 1502, an inverse scaling and weighting unit 1503, a frequency-time transform unit 1504, a lowpass filter unit 1505, and an audio output unit 1506. The datastream-input unit 1501, scaling and weighting unit 1503, transform unit 1504 and audio output unit 1506 operate in a similar manner to the equivalent processes of the first embodiment (Fig. 6).

- 10 The layered bitplane-decoding unit 1502 reconstructs coefficients in layer and bitplane order. At lower decoded bitrates coefficients can only be recovered within a limited bandwidth range, defined by the bandwidth limit of the last layer decoded. With reference to Fig. 16, this can cause nonlinear artifacts 1601 in the time-domain output following frequency-to-time transformation 1504 if the final encoded layer is not decoded, due to the missing high frequency coefficients 1602. When the time-frequency transform is an MDCT, the nonlinear artifacts 1601 will appear close to the bandwidth limit 1603 of the last decoded layer, and the frequency range across which errors appear will be a function of transform length and the shape of analysis-synthesis windows.

- Low-pass filtering the transform output with a lowpass filter unit 1505 shown in Fig. 15 can reduce the audibility of these errors. The lowpass filter response 1604, defined by a filter cutoff frequency and transition bandwidth, will tradeoff bandwidth against artifact attenuation. Ideally the filter cutoff frequency should track the bandwidth limit 1603 of the last decoded layer. If the decoded bitrate changes from frame to frame, as may occur if the coded datastream is received over a variable-bandwidth channel link, an adaptive filter should be used where the filter cutoff frequency can adapt to variations in the decoded bandwidth limit.

Layered coding schemes based on arithmetic coding and offering fine-grain scalability have previously been described by Park (supra), where arithmetic coding is used to identify newly-significant coefficient locations

within each bitplane scan. Conversely, the layered bitplane coding methods described in the embodiments below use runlength coding for the significance map stage of each bitplane scan.

- 5 Referring again to Fig. 14, the layered bitplane encoding unit 1405 codes a set of coefficients in each bitplane of each layer which is restricted to coefficient frequencies within the bandwidth limit of the layer. The operation of an example layered bitplane encoding process 1405 according to the fifth embodiment of the invention will now be described with
- 10 reference to Fig. 17. This process broadly follows the fixed-bandwidth bitplane encoding process of the second embodiment shown in Fig. 8, but embeds list scans within an outer layer loop in order to achieve a layered datastream structure.
- 15 With reference to Fig. 17, the layered bitplane encoding process begins at step s1701 by reordering transform coefficients to the LIC in a data-independent manner such that coefficients with the same frequency index are clustered together within the LIC. At step s1702 the LSC is initialised to an empty list. The most significant bitplane across all layers is determined
- 20 at s1703 by finding the largest transform coefficient magnitude within the frame, and a code representing the corresponding threshold level output to an output buffer. Each layer is then coded in turn, beginning with the base layer (s1704).
- 25 Each layer is associated with a bit allocation at step s1705, and a bandwidth limit at s1706 that increases with each layer coded. An example bitrate-bandwidth relationship for a 5-layer coder with a transform sampling frequency of 48 kHz and a 1024-sample frame length is shown in table 1:

Layer	Bitrate Range (kbit/s)	Layer bit allocation (kbit/s)	Layer bit allocation (bits)	Layer Bandwidth Limit (kHz)
-------	---------------------------	-------------------------------------	-----------------------------------	--------------------------------------

0 (base)	< 24.0	24.0	512	4.0
1	24.0 → 40.0	16.0	341	5.0
2	40.0 → 56.0	16.0	341	8.0
3	56.0 → 72.0	16.0	342	12.0
4	≥ 72.0	remainder for frame	remainder for frame	24.0

Table 1

At step s1707 coding for each layer begins with the most significant bitplane and continues through lower bitplanes until the bit allocation for the layer has been expended. For each bitplane, the LIC is scanned to the layer bandwidth limit at step s1708 using runlength codes to identify significant entries. If a fixed-wordlength Golomb-Rice runlength code is used for the LIC scan at s1708, then the runlength code wordlength can be optimised for each frame, or each layer, or each bitplane within each layer, and output as side information as appropriate. If an adaptive Golomb-Rice runlength code is used, then the wordlength is reset to a small value at the start of each bitplane within each layer (ie at the start of step s1708). End-of-run's in LIC scans are coded by outputting consecutive '0's until the layer bandwidth limit is passed. When a significant entry is found within step s1708 the runlength code and sign bit are output to the output buffer and the coefficient is moved to the LSC. Significant coefficients identified in earlier bitplanes are refined at the current bitplane level within the LSC scan at step s1709. LSC scans can be efficiently coded by outputting a single refinement bit for each list entry tested.

At step s1710 the threshold  $T$  corresponding to the bitplane level is halved, and if bits remain from the bit allocation for the current layer at step s1711, coding progresses to the next bitplane within the layer. If the bit allocation for the current layer has been used at step s1711, and the test at step s1713 indicates this layer is not the final layer, coding progresses to the next layer. Note that while the bit allocation test for the current layer is shown in Fig.

17 at the end of each bitplane scan at step s1711, coding progresses to the next layer if at any point within a bitplane scan the remaining bit allocation for the layer is zero.

- 5 For each layer coding begins at step s1707 at the most significant bitplane determined by the largest coefficient in the frame, irrespective of whether this bitplane contains any new significant coefficients within the bandwidth limit for the current layer. If one or more layers contain coefficients much larger than other layers, then the most significant bitplanes of the latter
- 10 layers will not contain new significant coefficients, and LIC scans for these 'empty' bitplanes will be wasteful of bits. Coding efficiency can be improved by outputting a 1-bit flag prior to each LIC scan (s1708), to indicate the presence or otherwise of newly-significant LIC entries within the current bitplane up to the bandwidth limit of the current layer. If newly-
- 15 significant entries exist then the flag is set to '1' and the LIC is scanned at step s1708. If no significant entries exist then a '0' is output and coding for the current bitplane progresses to the LSC scan at s1709. Note that these bitplane significance flags need not be used for all layers, or for all bitplanes within a layer. In practice good results are usually obtained when
- 20 flags are used for all layers except the base layer.

- As shown in Fig. 13, each enhancement layer can include significance and refinement data for coefficients that originate from any frequency region up to the bandwidth limit of the current layer, including bandwidth regions of
- 25 earlier layers. In order to maintain the correct coding order across layer boundaries, if a coefficient has been coded at the current bitplane depth in previous layers then it is not re-coded within the current layer. For example, the refinement stage at step s1709 only outputs refinement bits for significant coefficients identified in earlier bitplanes and not refined at the
- 30 current bitplane level within previous layers.

Referring again to Fig. 15, the layered bitplane decoding unit 1502 decodes a set of coefficients in each bitplane of each layer that is restricted to

coefficient frequencies within the bandwidth limit of the respective layer. The operation of a layered bitplane decoding process 1502 according to the fifth embodiment of the invention (not shown) mirrors the encoding algorithm shown in Fig. 17, and is similar to the fixed-bandwidth bitplane decoding process of the second embodiment shown in Fig. 10 except that bitplane list scans are embedded within an outer layer loop in order to decode a layered datastream structure.

Referring once again to Fig. 14, the layered bitplane encoding unit 1405 codes a set of coefficients in each bitplane of each layer that is restricted to coefficient frequencies within the bandwidth limit of the layer. The operation of an example layered bitplane encoding process 1405 by a sixth embodiment of the invention will now be described. This layered bitplane encoding method is similar to that of the fifth embodiment, except that an enhancement is made by extracting LIC coefficients to form a subsequence which for each bitplane significance scan within each layer is coded before the remaining LIC coefficients.

With reference to Fig. 18, the first step s1801 of the encoding algorithm reorders transform coefficients to a list of insignificant coefficients (LIC), where as before the reordering process groups together coefficients with the same frequency index within the LIC. At step s1802 a list of significant coefficients (LSC) is initialised to an empty list. The most significant bitplane for the whole frame is determined at s1803, and the corresponding threshold level  $T$  is coded and output to the output buffer. Each layer is then coded in turn, beginning with the base layer (s1804).

Each layer is associated with a bit allocation (step s1805) and bandwidth limit (s1806). Coding of each layer begins with the most significant bitplane for the frame (s1807), irrespective of whether this bitplane contains any new significant coefficients within the layer bandwidth limit, and subsequently bitplanes are coded in order of significance. For each bitplane, a significance map is coded in three stages s1808, s1809 and s1810, which

collectively identify coefficients with most-significant magnitude bits in the current bitplane. Also for each bitplane, a refinement stage s1811 outputs less significant bits of significant coefficients identified in earlier bitplanes and not yet refined at this bitplane level. The threshold  $T$  corresponding to the current bitplane is halved at s1812, and if bits remain from the bit allocation for the current layer (s1813), coding progresses to the next bitplane. When the bit allocation for the current layer has been used coding progresses to the next layer at s1814. When the bit allocation for the final layer has been used, coding for the current frame terminates and at step s1816 the contents of the output buffer is written to the datastream output unit 1406.

The criteria used to extract coefficients from the LIC to form a subsequence at s1808 is that the extracted coefficients should have a higher expected probability of becoming significant in the pending bitplane scan than those coefficients that remain in the LIC. Suitable contexts for selecting coefficients to form the subsequence include:

- coefficients that are frequency-domain neighbours to significant coefficients with the same time index
- for frames containing more than one transform block ( $B > 1$ ), coefficients that are time-domain neighbours to significant coefficients with the same frequency index
- the significant neighbour 'age', or the bitplane difference between the significant neighbour MSB bitplane and the current bitplane
- coefficients that have some harmonic relationship to significant coefficients.

Note that while coefficient extraction can in theory take place at any point(s) within a bitplane scan, in practice a convenient point at which to form the subsequence is at the start of each bitplane scan within each layer (as shown for s1808 in Fig. 18).

The subsequence is coded at s1809 by scanning for significant entries using runlength codes. A suitable runlength code is a fixed or adaptive Golomb

code. When a significant entry is found a runlength code and sign bit are output to the output buffer and the coefficient is moved from the subsequence to the LSC. Coding the subsequence at s1809 before the remaining LIC coefficients at step s1810 results in improved coding efficiency for those frames where the final bitplane scan is only partially completed.

Following subsequence coding, the LIC is scanned for significant entries at s1810, also using runlength codes. If a fixed Golomb-Rice runlength code is used where wordlength  $n$  is fixed for all bitplanes of all frames, an optimal value for  $n$  is selected which on average results in the most compact significance map code. Alternatively  $n$  can be optimised for each frame at a certain target bitrate and sent as side information at the start of the frame, or optimised for each layer and sent as side information at the start of the layer, or optimised for each bitplane and sent as side information at the start of the bitplane. If an adaptive Golomb-Rice code is used then  $n$  is reset to a small value at the beginning of each bitplane scan within each layer (ie beginning of step s1810).

Referring once again to Fig. 15, the layered bitplane decoding unit 1502 decodes a set of coefficients in each bitplane of each layer that is restricted to coefficient frequencies within the bandwidth limit of the respective layer. The operation of a layered bitplane decoding process 1502 according to the sixth embodiment of the invention (not shown) mirrors the encoding algorithm shown in Fig. 18. Hence for each bitplane within each layer, significance map decoding comprises the steps of subsequence formation using the same context rules used in the encoder, decoding subsequence runlength codes and sign bits, and decoding LIC runlength codes and sign bits.

Fig. 19 is a block diagram of an audio transcoding apparatus of a seventh embodiment of the invention, where a coded datastream input is transcoded to a bitplane-encoded scalable datastream output. The transcoding apparatus

comprises a datastream input unit 1901, a coefficient reconstruction unit 1902, a bitplane encoding unit 1903, and a scalable datastream output unit 1904.

- 5 Coded data frames representing encoded audio data are received by a datastream input unit 1901. The datastream input unit 1901 may be a storage device such as a hard disk, a RAM, and a CD-ROM, or an interface to a public telephone line, a radio line, a LAN or the like. The coded data may be of fixed-bitrate (non-scalable) format, such as provided by the
- 10 MPEG-2 AAC coding standard, and as such will consist of frames of quantised and entropy-coded frequency-domain coefficients. The datastream input unit 1901 also receives coded side information such as banded weight data for each frame.
- Quantised and entropy-coded data for each frame is input to the coefficient
- 15 reconstruction unit 1902, where frequency-domain coefficients are reconstructed from their coded representations. Reconstructed coefficients are then input to the bitplane encoding unit 1903, where coefficient bits of equal significance are grouped together into bitplanes, and each bitplane coded in order of significance. The bitplane encoding unit 1903 can use any
- 20 of the bitplane encoding algorithms described for previous embodiments, and can be of fixed-bandwidth or layered design.

The final stage of the transcoding process shown in Fig. 19 is to transmit to a memory or an external apparatus the bitplane encoded data output from

25 the bitplane encoding unit 1903, and also side information such as banded weight data, by means of a datastream output unit 1904. The datastream output unit 1904 may be a storage device such as a hard disk, a RAM, and a CD-ROM, or an interface to a public telephone line, a radio line, a LAN or the like.

30

The apparatus shown in Fig. 19 is useful for converting fixed-bitrate coded audio data to a scalable datastream format. Since at all stages coded data is

processed in the frequency domain and at no point transformed to the time domain, such an apparatus can be computationally efficient.

Fig. 20 is a block diagram of an audio transcoding apparatus also according to the seventh embodiment of the invention, where a bitplane-encoded scalable datastream input is transcoded to a datastream output. The transcoding apparatus comprises a scalable datastream input unit 2001, a bitplane decoding unit 2002, a coefficient quantisation and coding unit 2003, and a datastream output unit 2004.

10

Coded data frames representing bitplane-encoded audio data are received by a datastream input unit 2001. The datastream input unit 2001 may be a storage device such as a hard disk, a RAM, and a CD-ROM, or an interface to a public telephone line, a radio line, a LAN or the like. The datastream input unit 2001 also receives coded side information such as banded weight data for each frame.

20 Coded data is input to a bitplane decoding unit 2002 that reconstructs frequency-domain coefficients in bitplane order. The bitplane decoding unit 2002 can use any of the bitplane decoding algorithms described for previous embodiments of the invention, and can be of fixed-bandwidth or layered design, depending on the format of the datastream received by the datastream input unit 2001.

25 Reconstructed frequency-domain coefficients are then requantised and entropy coded in the coefficient quantisation and coding unit 2003, according to the format required by the datastream output unit 2004. The output data may be of fixed-bitrate (non-scalable) format, such as provided by the MPEG-2 AAC coding standard.

30

The final stage of the transcoding process shown in Fig. 20 is to transmit to a memory or an external apparatus the coded data output from the coefficient quantisation and coding unit 2003, and also side information

such as banded weight data, by means of a datastream output unit 2004. The datastream output unit 2004 may be a storage device such as a hard disk, a RAM, and a CD-ROM, or an interface to a public telephone line, a radio line, a LAN or the like.

5

The apparatus shown in Fig. 20 is useful for converting bitplane-encoded scalable audio data to a fixed-bitrate datastream format. Since at all stages coded data is processed in the frequency domain and at no point transformed to the time domain, such an apparatus can be computationally efficient.

10

The previous embodiments have described single-channel coding cases. However, in general audio signals possess more than one channel, and of particular interest is the two-channel stereo case. The coding techniques described above for single-channel signals can also be used to code stereo and other multi-channel signals.

15

A common method of representing stereo signals for audio coding is as m-s channel pairs, where the 'mid' signal is obtained by summing left and right stereo channels, and the 'side' signal is obtained by forming the difference between the left and right channels. Sum and difference operations can be performed either in the time or frequency domains. M-S signals can be coded using the fixed-bandwidth bitplane coding methods described above by initially coding the mid and side signals independently, but outputting coded bitplanes of equal significance to a datastream in interleaved m-s order. Because the mid signal is usually larger than the side signal, the first few bitplanes of an interleaved output will often contain mid signal information only.

20

25

An alternative arrangement may be preferred when layered coding is used. For 2-channel layered coding the base (first) layer may be coded as a single-channel signal for the best subjective performance at lower bitrates – hence the base layer consists of a bitplane-coded mid signal only. The second layer then adds stereo coding to the same bandwidth as the first

30

layer, hence the second layer consists of a bitplane-coded side signal only. Subsequent layers will consist of interleaved mid-side bitplanes each corresponding to a new coding bandwidth limit.

- 5 In general, this application is intended to cover any adaptations or variations of the present invention; in particular it will be realised that elements described in the embodiments may be replaced with equivalent elements fulfilling the same function.

## Claims

1. A method for encoding audio signals to a datastream, comprising the steps of:

5

(a) reordering frequency-domain coefficients representing the audio signal to a coefficient list, where the list order preserves the frequency order of coefficients and groups together coefficients with the same frequency index;

10

(b) quantising the coefficients and coding bits of equal significance together in bitplanes, where bitplanes are coded in order of significance beginning with the most significant bitplane, and coding of one or more bitplanes comprises the steps of:

15

(i) locating newly-significant coefficients with most-significant magnitude bit (MSB) positions within the current bitplane, by runlength coding positions of coefficient list entries whose magnitudes equal or exceed a predetermined threshold level corresponding to the current bitplane;

20

(ii) coding the signs of said newly-significant coefficients;

25

(iii) removing said newly-significant coefficients from the coefficient list.

(c) outputting coded bitplane data to the datastream.

2. A method according to claim 1, wherein the datastream comprises a base layer and a number of enhancement layers having predetermined bandwidth limits, and further characterised in that the coefficients corresponding to the base layer having a bandwidth limit are quantised and coded until a bit allocation is reached, and then the coefficients

30

corresponding to an enhancement layer having a bandwidth limit are quantised and coded until a bit allocation is reached, the quantisation and coding being repeated until all layers have been coded.

- 5     3.     A method according to claims 1 or 2, where at step (iii) newly-significant coefficient list entries are moved to a list of significant coefficients (LSC), and less-significant magnitude bit information for significant coefficients identified in earlier bitplanes is coded by coding corresponding LSC entries with respect to the current threshold level.
- 10     4.     A method according to any previous claim, where the frequency-domain coefficients are modified discrete cosine transform coefficients.
- 15     5.     A method according to claims 1 to 3, where the frequency-domain coefficients are wavelet packet transform coefficients.
- 20     6.     A method according to claims 1 to 5, wherein prior to step (a), frequency-domain coefficients are weighted in a frequency-dependent manner.
- 25     7.     A method according to claim 6, where weighting is performed with a set of banded weight values which are coded and output as side information to the datastream.
- 30     8.     A method according to claims 1 to 7, wherein coefficient list runlength coding in the step (i) is performed by Golomb codes.
9.     A method according to claim 8, where the Golomb parameter adapts, and corresponding parameter side information is coded and output to the datastream.
10.    A method according to claim 8, where the Golomb parameter adapts within a bitplane according to data previously coded within the bitplane.

11. A method according to claim 10, where the Golomb parameter is reset at the beginning of coding a bitplane.
12. A method according to claims 1 to 11, wherein coefficient list  
5 runlength coding in the step (i) is performed by reversible variable length codes.
13. A method according to claims 1 to 12, wherein coefficient list  
10 runlength coding in the step (i) is completed following the final significant list entry by coding repeated symbols until the end of the coefficient list is passed.
14. A method according to claims 1 to 13, where coding of one or more  
15 bitplanes at step (b) also includes the steps of:
- (iv) forming a subsequence from coefficient list entries, where the subsequence selection criteria are based on increased expected probability of significance within the current bitplane;
- 20 (v) locating newly-significant subsequence entries using runlength codes before locating newly-significant coefficients amongst the remaining coefficient list entries in step (i).
15. A method according to claim 14, where for step (iv) a new  
25 subsequence is formed at the beginning of coding a bitplane.
16. A method according to claims 14 or 15, where the contexts for selecting coefficient list entries to form a subsequence include any of the following:
- 30 (i) spectral proximity to significant coefficients with the same time index;  
(ii) temporal proximity to significant coefficients with the same frequency index;

- (iii) the bitplane differences between most-significant bit (MSB) bitplanes of significant neighbour coefficients and the current bitplane;
- (iv) spectral harmonic relationships with significant coefficients.

5    17.    A method for decoding a datastream representing an audio signal, comprising the steps of:

10    (a) initialising entries in a coefficient list to zero, where the list order preserves the frequency order of coefficients and groups together coefficients with the same frequency index;

15    (b) decoding bitplane data from the datastream in order of significance beginning with the most significant bitplane, where bitplane data corresponds to quantised coefficient bits of equal significance, and decoding of one or more bitplanes comprises the steps of:

20           (i) decoding runlength codes to locate newly-significant coefficient list entries which have most-significant magnitude bit (MSB) positions within the current bitplane;

      (ii) setting magnitudes of said newly-significant coefficient list entries to a predetermined threshold level corresponding to the current bitplane;

25        (iii) decoding the signs of said newly-significant coefficient list entries;

      (iv) removing said newly-significant entries from the coefficient list.

30    (c) reordering significant coefficients removed from the coefficient list to a set of frequency-domain output coefficients.

18. A method according to claim 17, where at step (iv) newly-significant coefficient list entries are moved to a list of significant coefficients (LSC), and less-significant magnitude bit information for significant coefficients identified in earlier bitplanes is decoded by decoding LSC refinement data  
5 with respect to the current threshold level.

19. A method according to claims 17 or 18, where the frequency-domain output coefficients are reconstructed modified discrete cosine transform coefficients.  
10

20. A method according to claims 17 or 18, where the frequency-domain output coefficients are reconstructed wavelet packet transform coefficients.

21. A method according to claims 17 to 20, wherein following step (c),  
15 reconstructed output coefficients are inverse weighted in a frequency-dependent manner.

22. A method according to claim 21, where inverse weighting is performed with a set of banded weight values which are decoded from side  
20 information in the datastream.

23. A method according to claims 17 to 22, wherein coefficient list runlength codes in the step (i) are Golomb codes.

24. A method according to claim 23, where the Golomb parameter adapts according to side information decoded from the datastream.  
25

25. A method according to claim 23, where the Golomb parameter adapts within a bitplane according to data previously decoded within the  
30 bitplane.

26. A method according to claim 25, where the Golomb parameter is reset at the beginning of decoding a bitplane.

27. A method according to claims 17 to 26, wherein coefficient list runlength codes in the step (i) are reversible variable length codes.
- 5 28. A method according to claims 17 to 27, wherein coefficient list runlength decoding in the step (i) is completed following the final significant list entry by decoding repeated symbols until the end of the coefficient list is passed.
- 10 29. A method according to claims 17 to 28, where decoding of one or more bitplanes at step (b) also includes the steps of:
- (v) forming a subsequence from coefficient list entries, where the subsequence selection criteria are based on increased expected probability  
15 of significance within the current bitplane;
- (vi) decoding runlength codes to locate newly-significant subsequence entries before locating newly-significant coefficients amongst the remaining coefficient list entries in step (i).  
20
30. A method according to claim 29, where for step (v) a new subsequence is formed at the beginning of decoding a bitplane.
- 25 31. A method according to claims 29 or 30, where the contexts for selecting coefficient list entries to form a subsequence include any of the following:
- (i) spectral proximity to significant coefficients with the same time index;  
(ii) temporal proximity to significant coefficients with the same frequency index;  
30 (iii) the bitplane differences between most-significant bit (MSB) bitplanes of significant neighbour coefficients and the current bitplane;  
(iv) spectral harmonic relationships with significant coefficients.

32. A method for encoding audio signals to a layered datastream having a base layer and a predetermined number of enhancement layers, comprising the steps of:
- 5 (a) reordering frequency-domain coefficients representing an audio signal to a coefficient list, where the list order preserves the frequency order of coefficients and groups together coefficients with the same frequency index;
  - 10 (b) quantising and coding coefficients corresponding to the base layer with a predetermined bandwidth limit, until a predetermined bit allocation for the base layer is reached;
  - (c) quantising and coding coefficients corresponding to the next  
15 enhancement layer with a predetermined bandwidth limit, until a predetermined bit allocation for the enhancement layer is reached;
  - (d) sequentially performing step (c) until all layers have been coded, wherein steps (b), (c) and (d) each includes coding quantised coefficient  
20 bits of equal significance together in bitplanes, where bitplanes are coded in order of significance beginning with the most significant bitplane, and coding of one or more bitplanes comprises the steps of:
    - 25 (i) locating newly-significant coefficients with most-significant magnitude bit (MSB) positions within the current bitplane, by runlength coding positions of coefficient list entries whose magnitudes equal or exceed a predetermined threshold level corresponding to the current bitplane;
    - 30 (ii) coding the signs of said newly-significant coefficients;
    - (iii) removing said newly-significant coefficients from the coefficient list.

(e) outputting coded layer data to the datastream.

33. A method according to claim 32, where coding of the next  
5 enhancement layer at step (c) also includes coding quantised coefficient  
data that is contained within previous layer bandwidth limits and that  
remains uncoded.

34. A method according to claims 32 or 33, where at step (iii) newly-  
10 significant coefficient list entries are moved to a list of significant  
coefficients (LSC), and less-significant magnitude bit information for  
significant coefficients identified in earlier bitplanes is coded by coding  
corresponding LSC entries with respect to the current threshold level.

15 35. A method according to claims 32 to 34, where the frequency-domain  
coefficients are modified discrete cosine transform coefficients.

36. A method according to claims 32 to 34, where the frequency-domain  
coefficients are wavelet packet transform coefficients.

20 37. A method according to claims 32 to 36, wherein prior to step (a),  
frequency-domain coefficients are weighted in a frequency-dependent  
manner.

25 38. A method according to claim 37, where weighting is performed with  
a set of banded weight values which are coded and output as side  
information to the datastream.

39. A method according to claims 32 to 38, where at the beginning of  
30 step (i) a flag preceding any runlength codes is coded to indicate whether  
the coefficient list contains any newly-significant coefficients within the  
bandwidth limit of the current layer.

40. A method according to claim 39, where the flag code is a single bit.
41. A method according to claims 32 to 40, wherein coefficient list runlength coding in the step (i) is performed by Golomb codes.
- 5 42. A method according to claim 41, where the Golomb parameter adapts, and corresponding parameter side information is coded and output to the datastream.
- 10 43. A method according to claim 41, where the Golomb parameter adapts within a bitplane according to data previously coded within the bitplane.
- 15 44. A method according to claim 43, where the Golomb parameter is reset at the beginning of coding a bitplane within a layer.
- 20 45. A method according to claims 32 to 44, wherein coefficient list runlength coding in the step (i) is performed by reversible variable length codes.
- 25 46. A method according to claims 32 to 45, wherein coefficient list runlength coding in the step (i) is completed following the final significant list entry by coding repeated symbols until the coefficient list bandwidth limit for the current layer is passed.
- 30 47. A method according to claims 32 to 46, where coding of one or more bitplanes within a layer at step (d) also includes the steps of:
- (iv) forming a subsequence from coefficient list entries, where the subsequence selection criteria are based on increased expected probability of significance within the current bitplane;

(v) locating newly-significant subsequence entries using runlength codes before locating newly-significant coefficients amongst the remaining coefficient list entries in step (i).

5 48. A method according to claim 47, where for step (iv) a new subsequence is formed at the beginning of coding a bitplane within a layer.

49. A method according to claims 47 or 48, where the contexts for selecting coefficient list entries to form a subsequence include any of the  
10 following:

- (i) spectral proximity to significant coefficients with the same time index;
- (ii) temporal proximity to significant coefficients with the same frequency index;
- (iii) the bitplane differences between most-significant bit (MSB) bitplanes  
15 of significant neighbour coefficients and the current bitplane;
- (iv) spectral harmonic relationships with significant coefficients.

50. A method for decoding audio signals from a layered datastream having a base layer and a predetermined number of enhancement layers,  
20 comprising the steps of:

(a) initialising entries in a coefficient list to zero, where the list order preserves the frequency order of coefficients and groups together coefficients with the same frequency index;  
25

(b) decoding data from the datastream corresponding to the base layer with a predetermined bandwidth limit, until a predetermined bit allocation for the base layer is reached;

30 (c) decoding data from the datastream corresponding to the next enhancement layer with a predetermined bandwidth limit, until a predetermined bit allocation for the enhancement layer is reached;

(d) sequentially performing step (c) until all layers have been decoded, wherein steps (b), (c) and (d) each includes decoding bitplane data corresponding to quantised coefficient bits of equal significance, where bitplanes are decoded in order of significance beginning with the most-significant bitplane, and decoding of one or more bitplanes comprises the steps of:

10 (i) decoding runlength codes to locate newly-significant coefficient list entries which have most-significant magnitude bit (MSB) positions within the current bitplane;

(ii) setting said newly-significant coefficient list entries to a predetermined threshold level corresponding to the current bitplane;

15 (iii) decoding the signs of said newly-significant coefficient list entries;

(iv) removing said newly-significant entries from the coefficient list.

20 (e) reordering significant coefficients removed from the coefficient list to a set of frequency-domain output coefficients.

51. A method according to claim 50, where decoding of data corresponding to the next enhancement layer at step (c) also includes  
25 decoding data for coefficients contained within previous layer bandwidth limits.

52. A method according to claims 50 or 51, where at step (iv) newly-significant coefficient list entries are moved to a list of significant  
30 coefficients (LSC), and less-significant magnitude bit information for significant coefficients identified in earlier bitplanes is decoded by decoding LSC refinement data with respect to the current threshold level.

53. A method according to claims 50 to 52, where the frequency-domain output coefficients are reconstructed modified discrete cosine transform coefficients.
- 5 54. A method according to claims 50 to 52, where the frequency-domain output coefficients are reconstructed wavelet packet transform coefficients.
55. A method according to claims 50 to 54, wherein following step (e), reconstructed output coefficients are inverse weighted in a frequency-  
10 dependent manner.
56. A method according to claim 55, where inverse weighting is performed with a set of banded weight values which are decoded from side information in the datastream.  
15
57. A method according to claims 50 to 56, where at the beginning of step (i) a flag preceding any runlength codes is decoded to indicate whether the coefficient list contains any newly-significant coefficients within the bandwidth limit of the current layer.  
20
58. A method according to claim 57, where the flag code is a single bit.
59. A method according to claims 50 to 58, wherein coefficient list runlength codes in the step (i) are Golomb codes.  
25
60. A method according to claim 59, where the Golomb parameter adapts according to side information decoded from the datastream.
61. A method according to claim 59, where the Golomb parameter adapts within a bitplane according to data previously decoded within the  
30 bitplane.

62. A method according to claim 61, where the Golomb parameter is reset at the beginning of decoding a bitplane within a layer.
63. A method according to claims 50 to 62, wherein coefficient list  
5 runlength codes in the step (i) are reversible variable length codes.
64. A method according to claims 50 to 63, wherein coefficient list runlength decoding in the step (i) is completed following the final significant list entry by decoding repeated symbols until the coefficient list  
10 bandwidth limit for the current layer is passed.
65. A method according to claims 50 to 64, where decoding of one or more bitplanes within a layer at step (d) also includes the steps of:
- 15 (v) forming a subsequence from coefficient list entries, where the subsequence selection criteria are based on increased expected probability of significance within the current bitplane;
- (vi) decoding runlength codes to locate newly-significant subsequence  
20 entries before locating newly-significant coefficients amongst the remaining coefficient list entries in step (i).
66. A method according to claim 65, where for step (v) a new subsequence is formed at the beginning of decoding a bitplane within a  
25 layer.
67. A method according to claims 65 or 66, where the contexts for selecting coefficient list entries to form a subsequence include any of the following:
- 30 (i) spectral proximity to significant coefficients with the same time index;  
(ii) temporal proximity to significant coefficients with the same frequency index;

- (iii) the bitplane differences between most-significant bit (MSB) bitplanes of significant neighbour coefficients and the current bitplane;
- (iv) spectral harmonic relationships with significant coefficients.

5

68. A method for decoding audio signals from a layered datastream having a base layer and a predetermined number of enhancement layers, where decoding of each frame of coded data comprises the steps of:

10 (a) decoding data from the datastream and reconstructing output coefficients corresponding to the base layer with a predetermined bandwidth limit, until a predetermined bit allocation for the base layer is reached or all of the data for the frame has been decoded;

15 (b) decoding data from the datastream and reconstructing output coefficients corresponding to the next enhancement layer with a predetermined bandwidth limit, until a predetermined bit allocation for the enhancement layer is reached or all of the data for the frame has been decoded;

20

(c) sequentially performing step (b) until all layers have been decoded, or until all of the data for the frame has been decoded;

25 (d) transforming reconstructed output coefficients to a time-domain output signal;

(e) lowpass filtering the time-domain output signal, where the lowpass filter cutoff frequency is dependent on the bandwidth limit of the last layer decoded.

30

69. A method according to claim 68, where decoding of data corresponding to the next enhancement layer at step (b) also includes

decoding data for coefficients contained within previous layer bandwidth limits.

5 70. A method according to claims 68 or 69, where the lowpass filter cutoff frequency is adapted in time.

71. A method according to any of the preceding claims, where the datastream is a bitstream.

10

72. An apparatus for encoding audio signals to a datastream, the apparatus comprising:

15 (a) reordering means for reordering frequency-domain coefficients representing an audio signal to a coefficient list, where the reordering means is configured to preserve the frequency order of coefficients within the list, and to group together coefficients with the same frequency index;

20 (b) bitplane coding means for quantising the coefficients and coding bits of equal significance together in bitplanes, where the bitplane coding means is configured to code bitplanes in order of significance beginning with the most-significant bitplane, and coding of one or more bitplanes comprises the steps of:

25 (i) locating newly-significant coefficients with most-significant magnitude bit (MSB) positions within the current bitplane, by runlength coding positions of coefficient list entries whose magnitudes equal or exceed a predetermined threshold level corresponding to the current bitplane;

30

(ii) coding the signs of said newly-significant coefficients;

(iii) removing said newly-significant coefficients from the coefficient list.

(c) means for outputting coded bitplane data to the datastream.

5

73. An apparatus according to claim 72, where at step (iii) means are also provided for moving newly-significant coefficient list entries to a list of significant coefficients (LSC), and for coding less-significant magnitude bit information for significant coefficients identified in earlier bitplanes by coding corresponding LSC entries with respect to a threshold level  
10 corresponding to the current bitplane.

15

74. An apparatus according to claims 72 or 73, where the frequency-domain coefficients are modified discrete cosine transform coefficients.

75. An apparatus according to claims 72 or 73, where the frequency-domain coefficients are wavelet packet transform coefficients.

20

76. An apparatus according to claims 72 to 75, where prior to step (a), weighting means is provided for weighting frequency-domain coefficients in a frequency-dependent manner.

25

77. An apparatus according to claim 76, wherein the weighting means is configured to perform weighting with a set of banded weight values which are coded and output as side information to the datastream.

30

78. An apparatus according to claims 72 to 77, wherein the bitplane coding means is configured so that coefficient list runlength coding in the step (i) is performed by Golomb codes.

79. An apparatus according to claim 78, where the Golomb parameter adapts, and corresponding parameter side information is coded and output to the datastream.

80. An apparatus according to claim 78, where the Golomb parameter adapts within a bitplane according to data previously coded within the bitplane.

5

81. An apparatus according to claim 80, where the Golomb parameter is reset at the beginning of coding a bitplane.

82. An apparatus according to claims 72 to 81, wherein the bitplane coding means is configured so that coefficient list runlength coding in the step (i) is performed by reversible variable length codes.

10

83. An apparatus according to claims 72 to 82, wherein the bitplane coding means is configured so that coefficient list runlength coding in the step (i) is completed following the final significant list entry by coding repeated symbols until the end of the coefficient list is passed.

15

84. An apparatus according to claims 72 to 83, wherein the bitplane coding means for one or more bitplanes at step (b) also includes the steps of:

20

(iv) forming a subsequence from coefficient list entries, where the subsequence selection criteria are based on increased expected probability of significance within the current bitplane;

25

(v) locating newly-significant subsequence entries using runlength codes before locating newly-significant coefficients amongst the remaining coefficient list entries in step (i).

85. An apparatus according to claim 84, where for step (iv) a new subsequence is formed at the beginning of coding a bitplane.

30

86. An apparatus according to claims 84 or 85, where the contexts for selecting coefficient list entries to form a subsequence include any of the following:

- (i) spectral proximity to significant coefficients with the same time index;
- 5 (ii) temporal proximity to significant coefficients with the same frequency index;
- (iii) the bitplane differences between most-significant bit (MSB) bitplanes of significant neighbour coefficients and the current bitplane;
- (iv) spectral harmonic relationships with significant coefficients.

10 87. An apparatus for decoding a datastream representing an audio signal, the apparatus comprising:

15 (a) means for initialising entries in a coefficient list to zero, where the list order preserves the frequency order of coefficients and groups together coefficients with the same frequency index;

20 (b) bitplane decoding means for decoding bitplane data from the datastream in order of significance beginning with the most significant bitplane, where bitplane data corresponds to quantised coefficient bits of equal significance, and decoding of one or more bitplanes comprises the steps of:

25 (i) decoding runlength codes to locate newly-significant coefficient list entries which have most-significant magnitude bit (MSB) positions within the current bitplane;

(ii) setting magnitudes of said newly-significant coefficient list entries to a predetermined threshold level corresponding to the current bitplane;

30 (iii) decoding the signs of said newly-significant coefficient list entries;

(iv) removing said newly-significant entries from the coefficient list.

(c) means for reordering significant coefficients removed from the coefficient list to a set of frequency-domain output coefficients.

5

88. An apparatus according to claim 87, where at step (iv) means are also provided for moving newly-significant coefficient list entries to a list of significant coefficients (LSC), and less-significant magnitude bit information for significant coefficients identified in earlier bitplanes is  
10 decoded by decoding LSC refinement data with respect to the current threshold level.

89. An apparatus according to claims 87 or 88, where the frequency-domain output coefficients are reconstructed modified discrete cosine  
15 transform coefficients.

90. An apparatus according to claims 87 or 88, where the frequency-domain output coefficients are reconstructed wavelet packet transform  
20 coefficients.

91. An apparatus according to claims 87 to 90, wherein following step (c), inverse weighting means is provided for inverse weighting reconstructed output coefficients in a frequency-dependent manner.

25 92. An apparatus according to claim 91, wherein the inverse weighting means is configured to perform inverse weighting with a set of banded weight values which are decoded from side information in the datastream.

30 93. An apparatus according to claims 87 to 92, wherein the bitplane decoding means is configured so that coefficient list runlength codes in the step (i) are Golomb codes.

94. An apparatus according to claim 93, where the Golomb parameter adapts according to side information decoded from the datastream.
95. An apparatus according to claim 93, where the Golomb parameter  
5 adapts within a bitplane according to data previously decoded within the bitplane.
96. An apparatus according to claim 95, where the Golomb parameter is reset at the beginning of decoding a bitplane.  
10
97. An apparatus according to claims 87 to 96, wherein the bitplane decoding means is configured so that coefficient list runlength codes in the step (i) are reversible variable length codes.
- 15 98. An apparatus according to claims 87 to 97, wherein the bitplane decoding means is configured so that coefficient list runlength decoding in the step (i) is completed following the final significant list entry by decoding repeated symbols until the end of the coefficient list is passed.
- 20 99. An apparatus according to claims 87 to 98, wherein the bitplane decoding means for one or more bitplanes at step (b) also includes the steps of:
- (v) forming a subsequence from coefficient list entries, where the  
25 subsequence selection criteria are based on increased expected probability of significance within the current bitplane;
- (vi) decoding runlength codes to locate newly-significant subsequence  
30 entries before locating newly-significant coefficients amongst the remaining coefficient list entries in step (i).
100. An apparatus according to claim 99, where for step (v) a new subsequence is formed at the beginning of decoding a bitplane.

101. An apparatus according to claims 99 or 100, where the contexts for selecting coefficient list entries to form a subsequence include any of the following:

- 5 (i) spectral proximity to significant coefficients with the same time index;
- (ii) temporal proximity to significant coefficients with the same frequency index;
- (iii) the bitplane differences between most-significant bit (MSB) bitplanes of significant neighbour coefficients and the current bitplane;
- 10 (iv) spectral harmonic relationships with significant coefficients.

102. An apparatus for encoding audio signals to a layered datastream having a base layer and a predetermined number of enhancement layers, the apparatus comprising:

- 15 (a) means for reordering frequency-domain coefficients representing an audio signal to a coefficient list, where the list order preserves the frequency order of coefficients and groups together coefficients with the same frequency index;
- 20 (b) means for quantising and coding coefficients corresponding to the base layer with a predetermined bandwidth limit, until a predetermined bit allocation for the base layer is reached;
- 25 (c) means for quantising and coding coefficients corresponding to the next enhancement layer with a predetermined bandwidth limit, until a predetermined bit allocation for the enhancement layer is reached;
- 30 (d) means for sequentially performing step (c) until all layers have been coded, wherein steps (b), (c) and (d) each includes bitplane coding means for coding quantised coefficient bits of equal significance together in bitplanes, where the bitplane coding means is configured to code bitplanes

in order of significance beginning with the most significant bitplane, and coding of one or more bitplanes comprises the steps of:

- 5           (i) locating newly-significant coefficients with most-significant magnitude bit (MSB) positions within the current bitplane, by runlength coding positions of coefficient list entries whose magnitudes equal or exceed a predetermined threshold level corresponding to the current bitplane;
- 10          (ii) coding the signs of said newly-significant coefficients;
- (iii) removing said newly-significant coefficients from the coefficient list.
- 15       (e) means for outputting coded layer data to the datastream.

103. An apparatus according to claim 102, where at step (c) means are also provided for coding of the next enhancement layer to include coding quantised coefficient data that is contained within previous layer bandwidth  
20 limits and that remains uncoded.

104. An apparatus according to claims 102 or 103, where at step (iii) means are also provided for moving newly-significant coefficient list entries to a list of significant coefficients (LSC), and less-significant  
25 magnitude bit information for significant coefficients identified in earlier bitplanes is coded by coding corresponding LSC entries with respect to the current threshold level.

105. An apparatus according to claims 102 to 104, where the frequency-  
30 domain coefficients are modified discrete cosine transform coefficients.

106. An apparatus according to claims 102 to 104, where the frequency-domain coefficients are wavelet packet transform coefficients.

107. An apparatus according to claims 102 to 106, wherein prior to step (a), weighting means is provided for weighting frequency-domain coefficients in a frequency-dependent manner.

5

108. An apparatus according to claim 107, wherein the weighting means is configured to perform weighting with a set of banded weight values which are coded and output as side information to the datastream.

10 109. An apparatus according to claims 102 to 108, wherein the bitplane coding means is configured so that at the beginning of step (i) a flag preceding any runlength codes is coded to indicate whether the coefficient list contains any newly-significant coefficients within the bandwidth limit of the current layer.

15

110. An apparatus according to claim 109, where the flag code is a single bit.

20 111. An apparatus according to claims 102 to 110, wherein the bitplane coding means is configured so that coefficient list runlength coding in the step (i) is performed by Golomb codes.

25 112. An apparatus according to claim 111, where the Golomb parameter adapts, and corresponding parameter side information is coded and output to the datastream.

30 113. An apparatus according to claim 111, where the Golomb parameter adapts within a bitplane according to data previously coded within the bitplane.

114. An apparatus according to claim 113, where the Golomb parameter is reset at the beginning of coding a bitplane within a layer.

115. An apparatus according to claims 102 to 114, wherein the bitplane coding means is configured so that coefficient list runlength coding in the step (i) is performed by reversible variable length codes.

5 116. An apparatus according to claims 102 to 115, wherein the bitplane coding means is configured so that coefficient list runlength coding in the step (i) is completed following the final significant list entry by coding repeated symbols until the coefficient list bandwidth limit for the current layer is passed.

10

117. An apparatus according to claims 102 to 116, wherein the bitplane coding means for one or more bitplanes within a layer at step (d) also includes the steps of:

15 (iv) forming a subsequence from coefficient list entries, where the subsequence selection criteria are based on increased expected probability of significance within the current bitplane;

20 (v) locating newly-significant subsequence entries using runlength codes before locating newly-significant coefficients amongst the remaining coefficient list entries in step (i).

25 118. An apparatus according to claim 117, where for step (iv) a new subsequence is formed at the beginning of coding a bitplane within a layer.

25

119. An apparatus according to claims 117 or 118, where the contexts for selecting coefficient list entries to form a subsequence include any of the following:

- 30 (i) spectral proximity to significant coefficients with the same time index;  
(ii) temporal proximity to significant coefficients with the same frequency index;  
(iii) the bitplane differences between most-significant bit (MSB) bitplanes of significant neighbour coefficients and the current bitplane;

(iv) spectral harmonic relationships with significant coefficients.

120. An apparatus for decoding audio signals from a layered datastream having a base layer and a predetermined number of enhancement layers, the  
5 apparatus comprising:

(a) means for initialising entries in a coefficient list to zero, where the list order preserves the frequency order of coefficients and groups together coefficients with the same frequency index;

10

(b) means for decoding data from the datastream corresponding to the base layer with a predetermined bandwidth limit, until a predetermined bit allocation for the base layer is reached;

15 (c) means for decoding data from the datastream corresponding to the next enhancement layer with a predetermined bandwidth limit, until a predetermined bit allocation for the enhancement layer is reached;

20 (d) means for sequentially performing step (c) until all layers have been decoded, wherein steps (b), (c) and (d) each includes bitplane decoding means for decoding bitplane data corresponding to quantised coefficient bits of equal significance, where bitplanes are decoded in order of significance beginning with the most-significant bitplane, and decoding of one or more bitplanes comprises the steps of:

25

(i) decoding runlength codes to locate newly-significant coefficient list entries which have most-significant magnitude bit (MSB) positions within the current bitplane;

30

(ii) setting said newly-significant coefficient list entries to a predetermined threshold level corresponding to the current bitplane;

(iii) decoding the signs of said newly-significant coefficient list entries;

(iv) removing said newly-significant entries from the coefficient list.

5

(e) means for reordering significant coefficients removed from the coefficient list to a set of frequency-domain output coefficients.

10 121. An apparatus according to claim 120, where at step (c) means are provided for decoding of data corresponding to the next enhancement layer to also include decoding data for coefficients contained within previous layer bandwidth limits.

15 122. An apparatus according to claims 120 or 121, where at step (iv) means are also provided for moving newly-significant coefficient list entries to a list of significant coefficients (LSC), and less-significant magnitude bit information for significant coefficients identified in earlier bitplanes is decoded by decoding LSC refinement data with respect to the current threshold level.

20

123. An apparatus according to claims 120 to 122, where the frequency-domain output coefficients are reconstructed modified discrete cosine transform coefficients.

25 124. An apparatus according to claims 120 to 122, where the frequency-domain output coefficients are reconstructed wavelet packet transform coefficients.

30 125. An apparatus according to claims 120 to 124, wherein following step (e), inverse weighting means is provided for inverse weighting frequency-domain output coefficients in a frequency-dependent manner.

126. An apparatus according to claim 125, wherein the inverse weighting means is configured to perform inverse weighting with a set of banded weight values which are decoded from side information in the datastream.
- 5 127. An apparatus according to claims 120 to 126, wherein the bitplane decoding means is configured so that at the beginning of step (i) a flag preceding any runlength codes is decoded to indicate whether the coefficient list contains any newly-significant coefficients within the bandwidth limit of the current layer.
- 10 128. An apparatus according to claim 127, where the flag code is a single bit.
- 15 129. An apparatus according to claims 120 to 128, wherein the bitplane decoding means is configured so that coefficient list runlength codes in the step (i) are Golomb codes.
- 20 130. An apparatus according to claim 129, where the Golomb parameter adapts according to side information decoded from the datastream.
- 25 131. An apparatus according to claim 129, where the Golomb parameter adapts within a bitplane according to data previously decoded within the bitplane.
- 30 132. An apparatus according to claim 131, where the Golomb parameter is reset at the beginning of decoding a bitplane within a layer.
133. An apparatus according to claims 120 to 132, wherein the bitplane decoding means is configured so that coefficient list runlength codes in the step (i) are reversible variable length codes.
134. An apparatus according to claims 120 to 133, wherein the bitplane decoding means is configured so that coefficient list runlength decoding in

the step (i) is completed following the final significant list entry by decoding repeated symbols until the coefficient list bandwidth limit for the current layer is passed.

- 5 135. An apparatus according to claims 120 to 134, wherein the bitplane decoding means for one or more bitplanes within a layer at step (d) also includes the steps of:

10 (v) forming a subsequence from coefficient list entries, where the subsequence selection criteria are based on increased expected probability of significance within the current bitplane;

15 (vi) decoding runlength codes to locate newly-significant subsequence entries before locating newly-significant coefficients amongst the remaining coefficient list entries in step (i).

20 136. An apparatus according to claim 135, where for step (v) a new subsequence is formed at the beginning of decoding a bitplane within a layer.

25 137. An apparatus according to claims 135 or 136, where the contexts for selecting coefficient list entries to form a subsequence include any of the following:

- (i) spectral proximity to significant coefficients with the same time index;  
30 (ii) temporal proximity to significant coefficients with the same frequency index;  
(iii) the bitplane differences between most-significant bit (MSB) bitplanes of significant neighbour coefficients and the current bitplane;  
(iv) spectral harmonic relationships with significant coefficients.

138. An apparatus for decoding audio signals from a layered datastream having a base layer and a predetermined number of enhancement layers, the apparatus for decoding each frame of coded data comprising:

(a) means for decoding data from the datastream and reconstructing output coefficients corresponding to the base layer with a predetermined bandwidth limit, until a predetermined bit allocation for the base layer is reached or all of the data for the frame has been decoded;

(b) means for decoding data from the datastream and reconstructing output coefficients corresponding to the next enhancement layer with a predetermined bandwidth limit, until a predetermined bit allocation for the enhancement layer is reached or all of the data for the frame has been decoded;

(c) means for sequentially performing step (b) until all layers have been decoded, or until all of the data for the frame has been decoded;

(d) means for transforming reconstructed output coefficients to a time-domain output signal;

(e) filter means for lowpass filtering the time-domain output signal, where the filter means is configured so that the lowpass filter cutoff frequency is dependent on the bandwidth limit of the last layer decoded.

139. An apparatus according to claim 138, where at step (b) means are also provided for decoding data corresponding to the next enhancement layer which includes data for coefficients contained within previous layer bandwidth limits.

140. An apparatus according to claims 138 or 139, wherein the filter means is configured so that the lowpass filter cutoff frequency is adapted in time.

141. An apparatus according to claims 72 to 140, where the datastream is a bitstream.



Application No: GB 0210704.3  
Claims searched: 1-67,71-137,141

Examiner: Owen Wheeler  
Date of search: 25 November 2002

**Patents Act 1977**  
**Search Report under Section 17**

**Databases searched:**

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.T): H4F (FRR, FRT) H4P (PDCFR, PDCFX) H4R (RPBE)

Int Cl (Ed.7): G06F: 17/14; H03M: 7/46; H04B: 1/66; H04L: 29/06;  
H04N: 1/41, 5/926, 7/30, 7/50, 9/804

Other: Online: EPODOC, JAPIO, WPI

**Documents considered to be relevant:**

Category	Identity of document and relevant passage	Relevant to claims
A	EP 0884850 A2 [SAMSUNG]	
A	WO 2001/062009 [SIEMENS]	
A	WO 1998/037700 [BRITISH TECHNOLOGY GROUP LTD]	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.